

**PENERAPAN METODE *USER CENTERED DESIGN* PADA
APLIKASI *E-LEARNING* BERBASIS ANDROID
(STUDI KASUS: SMAN 3 SIDOARJO)**

SKRIPSI

Untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun oleh:
Arie Krisnoanto
NIM: 135150201111068



PROGRAM STUDI TEKNIK INFORMATIKA
JURUSAN TEKNIK INFORMATIKA
FAKULTAS ILMU KOMPUTER
UNIVERSITAS BRAWIJAYA
MALANG
2018

PENGESAHAN

PENERAPAN METODE *USER CENTERED DESIGN* PADA APLIKASI *E-LEARNING*
BERBASIS ANDROID (STUDI KASUS: SMAN 3 SIDOARJO)

SKRIPSI

Diajukan untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun Oleh :
Arie Krisnoanto
NIM: 135150201111068

Skripsi ini telah diuji dan dinyatakan lulus pada
26 Juli 2018

Telah diperiksa dan disetujui oleh:

Dosen Pembimbing I



Adam Hendra Brata, S.Kom., M.T., M.Sc.
NIK: 2016079001051001

Dosen Pembimbing II



Mahardeka Tri Ananta, S.Kom., M.T., M.Sc.
NIK: 2016078912041001

Mengetahui
Ketua Jurusan Teknik Informatika



Tri Astoto Kurniawan, S.T, M.T, Ph.D
NIP: 19710518 200312 1 001

PERNYATAAN ORISINALITAS

Saya menyatakan dengan sebenar-benarnya bahwa sepanjang pengetahuan saya, di dalam naskah skripsi ini tidak terdapat karya ilmiah yang pernah diajukan oleh orang lain untuk memperoleh gelar akademik di suatu perguruan tinggi, dan tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis disitasi dalam naskah ini dan disebutkan dalam daftar pustaka.

Apabila ternyata didalam naskah skripsi ini dapat dibuktikan terdapat unsur-unsur plagiasi, saya bersedia skripsi ini digugurkan dan gelar akademik yang telah saya peroleh (sarjana) dibatalkan, serta diproses sesuai dengan peraturan perundang-undangan yang berlaku (UU No. 20 Tahun 2003, Pasal 25 ayat 2 dan Pasal 70).

Malang, 01 Agustus 2018



Arie Kristianto

NIM: 135150201111068

KATA PENGANTAR

Puji syukur penulis persembahkan pada hadirat Allah SWT atas rahmat-Nya yang telah diberikan sehingga penulis dapat menyelesaikan tugas akhir ini yang berjudul **“Penerapan Metode *User Centered Design* pada Aplikasi *E-Learning* Berbasis Android (Studi Kasus: SMAN 3 Sidoarjo)”**. Shalawat serta salam tercurahkan kepada Nabi Muhammad SAW, keluarga, sahabat beserta pengikutnya hingga akhir zaman.

Dalam penyelesaian tugas akhir ini, penulis juga mendapat banyak sekali bantuan, masukan dan motivasi dari berbagai pihak. Oleh karena itu dalam kesempatan ini penulis ingin menyampaikan terima kasih sebesar-besarnya kepada yang terhormat:

1. Kedua orang tua, papa Prasetyo dan mama Dewi Widiati, kakak Rina Rahmawati, S.Kom. dan segenap keluarga penulis yang telah memberikan *support* baik moril serta doa yang tidak putus selama masa penelitian berlangsung kepada penulis.
2. Bapak Wayan Firdaus Mahmudy, S.Si, M.T., Ph.D. selaku Dekan Fakultas Ilmu Komputer Universitas Brawijaya Malang.
3. Bapak Tri Astoto Kurniawan, S.T, M.T., Ph.D. selaku Ketua Jurusan Teknik Informatika Fakultas Ilmu Komputer Universitas Brawijaya Malang.
4. Bapak Agus Wahyu Widodo, S.Si, M.Sc. selaku Ketua Program Studi Teknik Informatika Fakultas Ilmu Komputer Universitas Brawijaya.
5. Bapak Adam Hendra Brata, S.Kom., M.T., M.Sc. dan Bapak Mahardeka Tri Ananta, S.Kom., M.T., M.Sc. selaku Dosen Pembimbing yang telah banyak membantu dan mengarahkan dalam proses penulisan skripsi ini.
6. Seluruh dosen Fakultas Ilmu Komputer Universitas Brawijaya yang telah memberikan ilmu kepada penulis.
7. Kerabat penulis Artiyana Prasetya, Diah Shinta Dewi, Lia Rosalina, Arya Yudha Mahendra, Meita Putri Aidha, Irfan Noor Agdhian, Anandhi Tristiaratri, S.Kom., Sifasani Q. Fauzia, S.Kom., Amalia Kartika Ariyani, S.Kom., dan Rusjda Widya Rahmani yang telah membantu tugas akhir saya dalam bentuk apapun.
8. Kepala sekolah, pengajar serta siswa-siswi SMAN Negeri 3 Sidoarjo. Annisa Nur Aini, salah satu siswi SMAN 3 Sidoarjo yang saat ini telah menjadi alumni (2018) bersedia membantu agar komunikasi dengan pihak sekolah tetap terjalin.
9. SABEB Entertainment, rekan organisasi BIOS (Badan Internal Olahraga dan Seni) dan rekan organisasi lainnya, Seluruh KBM FILKOM.

Penulis menyadari bahwa dalam proses penulisan skripsi ini masih terdapat kesalahan dan kekurangan di beberapa bagian. Karena itu kritikan dan saran yang membangun sangat penulis harapkan demi perbaikan dan pengembangan skripsi ini dimasa mendatang. Semoga skripsi ini bermanfaat bagi siapapun yang membaca dan menerapkannya dalam kehidupan sehari-hari.

Malang, 26 Juli 2018

Penulis

ariekrisnoanto@gmail.com



ABSTRAK

Pendidikan dan teknologi merupakan dua hal yang saling ketergantungan untuk kehidupan manusia dimasa depan. SMAN 3 Sidoarjo merupakan salah satu sekolah yang memiliki fasilitas edukasi untuk siswa dan pengajar yaitu *e-learning*, fasilitas yang sudah berjalan beberapa tahun di situs resmi sekolah ini masih dalam tahap pengembangan dan masih belum digunakan sepenuhnya dengan baik oleh pihak sekolah karena kurangnya minat penggunaan *e-learning* dari pihak siswa dan pengajar. Data dari statista.com terdapat 88,37% pengguna *smartphone* dengan sistem operasi Android hingga akhir tahun 2017. Dengan melihat dan memanfaatkan kondisi di lapangan, siswa dan pengajar telah menggunakan *smartphone* dalam kegiatan sehari-hari serta memaksimalkan penggunaan *e-learning* sekolah, aplikasi penunjang kegiatan pembelajaran atau *e-learning* sekolah diimplementasikan pada *smartphone* Android dengan menggunakan metode *User-Centered Design* yang mengacu pada *experience* pengguna sehingga dapat membuat aplikasi *e-learning* ini memiliki nilai *usability* yang cukup baik untuk digunakan oleh siswa dan pengajar. Disamping itu metode UCD bersifat iteratif yang artinya dapat menggali kebutuhan pengguna *end-user* yang dapat berubah-ubah secara langsung. Data uji yang dihasilkan saat menggunakan metode kuesioner USE terdapat hasil rata-rata kelayakan pada kriteria *usefulness* sebesar 91,71%, kriteria *ease of use* sebesar 83,37%, kriteria *ease of learning* sebesar 94,5%, dan kriteria *satisfaction* sebesar 82,16%. Untuk rata-rata hasil nilai uji *usability* menggunakan *task scenario* didapati 96,7% keberhasilan pada pengguna siswa dan 94,6% keberhasilan pada pengguna pengajar.

Kata kunci: *e-learning*, *ucd*, *usability*, *android*, *notification*

ABSTRACT

Education and technology are two interdependence thing for human life in the future. Senior high school 3 Sidoarjo is one of school that have e-learning as educational facilities for students and teachers. Facilities that have been running for several years on the official site of this school is still in the development stage and still not used completely well by the school due to lack of interest for using e-learning from students and teachers. Data from statista.com there are 88.37% smartphone Android users until the end of 2017. By viewing and utilizing the current conditions, students and teachers have been using smartphones in their daily activities as well as maximizing the use of school facilities, e-learning implemented on Android smartphones using User-Centered Design method that refers to the user experience to make e-learning application has a good usability value to be used by students and teachers. In addition, UCD method is iterative which means it can explore any needs of end-user that can change directly. The result of data test from USE questionnaire method are usefulness criteria 91.71%, ease of use criteria 83.37%, ease of learning criteria 94.5%, and satisfaction criteria 82,16%. The average score of usability testing using task scenarios had 96.7% success in student user and 94,6% success on teacher user.

keywords: e-learning, ucd, usability, android, notification

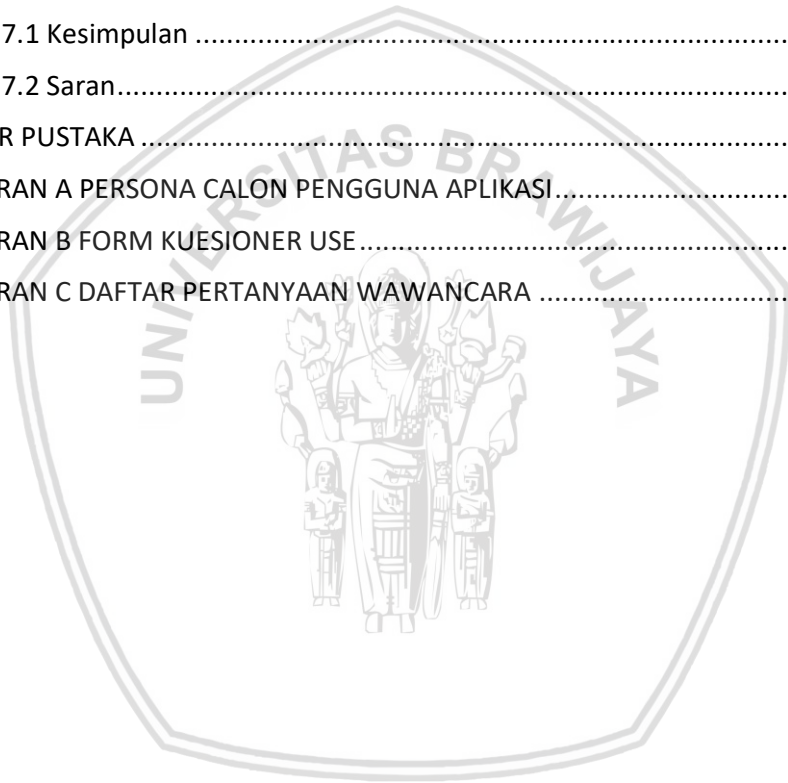
DAFTAR ISI

PENGESAHAN	ii
PERNYATAAN ORISINALITAS	iii
KATA PENGANTAR	iv
ABSTRAK.....	vi
ABSTRACT	vii
DAFTAR ISI	viii
DAFTAR TABEL	xii
DAFTAR GAMBAR	xv
DAFTAR LAMPIRAN	xx
BAB 1 PENDAHULUAN	1
1.1 Latar belakang.....	1
1.2 Rumusan masalah	3
1.3 Tujuan	3
1.4 Manfaat	3
1.5 Batasan masalah	3
1.6 Sistematika pembahasan.....	4
BAB 2 LANDASAN KEPUSTAKAAN.....	5
2.1 Kajian Pustaka	5
2.2 E-Learning	6
2.2.1 Konsep <i>E-Learning</i>	6
2.2.2 Komponen <i>E-Learning</i>	7
2.3 Android	7
2.3.1 Jenis Android	8
2.4 Firebase	9
2.5 OneSignal	11
2.6 SQLite	11
2.7 <i>User Centered Design</i> (UCD).....	12
2.7.2 Prinsip <i>User Centered Design</i>	13
2.7.3 Target <i>User Centered Design</i>	14
2.8 Pengujian Perangkat Lunak	14

2.8.1 Pengujian Unit	14
2.8.2 Pengujian Integrasi	14
2.8.3 Pengujian Validasi	14
2.8.4 Pengujian <i>Usability</i>	15
2.8.5 Pengujian <i>Compatibility</i>	17
BAB 3 METODOLOGI	18
3.1 Studi Literatur	18
3.2 Analisis Kebutuhan	19
3.3 Perancangan Perangkat Lunak	19
3.4 Implementasi Perangkat Lunak	20
3.5 Pengujian Perangkat Lunak	20
3.6 Pengambilan Kesimpulan dan Saran	21
BAB 4 REKASAYA KEBUTUHAN	22
4.1 Analisis Kebutuhan	22
4.1.1 Gambaran Umum Aplikasi	22
4.1.2 Identifikasi Aktor	23
4.2 Analisis Data	23
4.3 Analisis Kebutuhan Fungsional	24
4.4 Analisis Kebutuhan Non-Fungsional	30
4.5 Diagram <i>Use Case</i>	30
4.6 Diagram <i>Use Case</i> (Iterasi 1)	32
4.7 Diagram <i>Use Case</i> (Iterasi 2)	33
4.8 Skenario <i>Use Case</i>	34
4.8.1 Skenario <i>Use Case</i> Masuk (Siswa)	34
4.8.2 Skenario <i>Use Case</i> Masuk (Pengajar)	34
4.8.3 Skenario <i>Use Case</i> Lihat <i>Timeline</i> Kegiatan Akademik	35
4.8.4 Skenario <i>Use Case</i> Tambah Kegiatan Akademik	35
4.8.5 Skenario <i>Use Case</i> Lihat Jadwal Pelajaran	36
4.8.6 Skenario <i>Use Case</i> Lihat Info Perubahan Jadwal Pelajaran	37
4.8.7 Skenario <i>Use Case</i> Tambah Info Perubahan Jadwal Pelajaran ...	37
4.8.8 Skenario <i>Use Case</i> Lihat Materi Pelajaran	38
4.8.9 Skenario <i>Use Case</i> Lihat Aspirasi	38

4.8.10 Skenario <i>Use Case</i> Kirim Aspirasi	39
4.8.11 Skenario <i>Use Case</i> Lihat Tugas	39
4.8.12 Skenario <i>Use Case</i> Unggah Tugas	40
4.8.13 Skenario <i>Use Case</i> Tambah Tugas	40
4.8.14 Skenario <i>Use Case</i> Lihat Berkas Tugas	41
4.8.15 Skenario <i>Use Case</i> Unduh Berkas Tugas	41
4.8.16 Skenario <i>Use Case</i> Lihat Detil Akun	42
4.8.17 Skenario <i>Use Case</i> Ubah Detil Akun.....	42
4.8.18 Skenario <i>Use Case</i> Ubah <i>Password</i>	43
4.8.19 Skenario <i>Use Case</i> Pendaftaran.....	43
4.8.20 Skenario <i>Use Case</i> Keluar	44
4.8.21 Skenario <i>Use Case</i> Keluar	44
BAB 5 PERANCANGAN DAN IMPLEMETASI	46
5.1 Perancangan Sistem	46
5.1.1 Perancangan Arsitektur Sistem	46
5.1.2 Perancangan Basis Data	47
5.1.3 Perancangan <i>Class Diagram</i>	53
5.1.4 Perancangan <i>Sequence Diagram</i>	64
5.1.5 Perancangan Algoritme.....	66
5.1.6 Perancangan Antarmuka.....	68
5.2 Implementasi	89
5.2.1 Spesifikasi Sistem.....	89
5.2.2 Batasan Implementasi	90
5.2.3 Implementasi Basis Data	90
5.2.4 Implementasi <i>Class</i>	93
5.2.5 Implementasi Kode Program.....	96
5.2.6 Implementasi Antarmuka	99
BAB 6 PENGUJIAN	116
6.1 Pengujian Fungsional	116
6.1.1 Pengujian Unit	116
6.1.2 Pengujian Validasi	126
6.2 Pengujian Non-Fungsional.....	150

6.2.1 Pengujian <i>Usability</i>	150
6.2.2 Pengujian <i>Compatibility</i>	157
6.3 Analisis Hasil Pengujian	157
6.3.1 Analisis Hasil Pengujian Unit	157
6.3.2 Analisis Hasil Pengujian Validasi	158
6.3.3 Analisis Hasil Pengujian <i>Usability</i>	158
6.3.4 Analisis Hasil Pengujian <i>Compatibility</i>	161
BAB 7 PENUTUP	162
7.1 Kesimpulan	162
7.2 Saran	163
DAFTAR PUSTAKA	165
LAMPIRAN A PERSONA CALON PENGGUNA APLIKASI	167
LAMPIRAN B FORM KUESIONER USE	170
LAMPIRAN C DAFTAR PERTANYAAN WAWANCARA	172



DAFTAR TABEL

Tabel 2.1 Tabel Kuesioner USE	16
Tabel 2.2 Tabel Skala Likert	17
Tabel 4.1 Identifikasi Aktor	23
Tabel 4.2 Rekap Kebutuhan Pengguna (Siswa)	24
Tabel 4.3 Rekap Kebutuhan Pengguna (Pengajar)	24
Tabel 4.4 Kebutuhan Fungsional Siswa	25
Tabel 4.5 Kebutuhan Fungsional Siswa (Iterasi 1)	26
Tabel 4.6 Kebutuhan Fungsional Siswa (Iterasi 2)	27
Tabel 4.7 Kebutuhan Fungsional Pengajar	27
Tabel 4.8 Kebutuhan Fungsional Pengajar (Iterasi 1)	29
Tabel 4.9 Kebutuhan Fungsional Pengajar (Iterasi 2)	29
Tabel 4.10 Daftar Kebutuhan non-Fungsional	30
Tabel 4.11 Skenario <i>Use Case</i> Masuk	34
Tabel 4.12 Skenario <i>Use Case</i> Masuk	34
Tabel 4.13 Skenario <i>Use Case</i> Lihat Timeline Akademik	35
Tabel 4.14 Skenario <i>Use Case</i> Tambah Kegiatan Akademik	36
Tabel 4.15 Skenario <i>Use Case</i> Lihat Jadwal Pelajaran	36
Tabel 4.16 Skenario <i>Use Case</i> Lihat Info Perubahan Jadwal Pelajaran	37
Tabel 4.17 Skenario <i>Use Case</i> Tambah Info Perubahan Jadwal Pelajaran	37
Tabel 4.18 Skenario <i>Use Case</i> Lihat Materi Pelajaran	38
Tabel 4.19 Skenario <i>Use Case</i> Lihat Aspirasi	38
Tabel 4.20 Skenario <i>Use Case</i> Kirim Aspirasi	39
Tabel 4.21 Skenario <i>Use Case</i> Lihat Tugas	39
Tabel 4.22 Skenario <i>Use Case</i> Unggah Tugas	40
Tabel 4.23 Skenario <i>Use Case</i> Tambah Tugas	40
Tabel 4.24 Skenario <i>Use Case</i> Lihat Berkas Tugas	41
Tabel 4.25 Skenario <i>Use Case</i> Unduh Berkas Tugas	41
Tabel 4.26 Skenario <i>Use Case</i> Lihat Detil Akun	42
Tabel 4.27 Skenario <i>Use Case</i> Ubah Detil Akun	42
Tabel 4.28 Skenario <i>Use Case</i> Ubah Detil Akun	43

Tabel 4.29 Skenario <i>Use Case</i> Pendaftaran	43
Tabel 4.30 Skenario <i>Use Case</i> Keluar	44
Tabel 4.31 Skenario <i>Use Case</i> Keluar	44
Tabel 5.1 Tabel Keterangan Skema Basis Data Akun Siswa	48
Tabel 5.2 Tabel Keterangan Skema Basis Data Aspirasi	49
Tabel 5.3 Tabel Keterangan Skema Basis Data Aspirasi	49
Tabel 5.4 Tabel Keterangan Skema Basis Data Timeline Akademik	50
Tabel 5.5 Tabel Keterangan Skema Basis Data Berkas Tugas	51
Tabel 5.6 Tabel Keterangan Skema Basis Data Tugas Siswa	52
Tabel 5.7 Tabel Keterangan Skema Basis Data Tugas Siswa	53
Tabel 5.8 Spesifikasi Perangkat Keras	89
Tabel 5.9 Spesifikasi Perangkat Lunak	89
Tabel 5.10 Spesifikasi Perangkat Bergerak	90
Tabel 5.11 Implementasi <i>Class</i> Pada Kode Program Java	93
Tabel 5.12 Penjelasan <i>Method</i> tambahTugas()	97
Tabel 5.13 Penjelasan <i>Method</i> uploadFile()	98
Tabel 5.14 Penjelasan <i>Method</i> addAspirasi()	99
Tabel 6.1 Kasus Uji Pengujian Unit <i>method</i> getPDF()	118
Tabel 6.2 Kasus Uji Pengujian Unit <i>method</i> uploadFile()	120
Tabel 6.3 Kasus Uji Pengujian Unit <i>method</i> updatePertanyaan()	122
Tabel 6.4 Kasus Uji Pengujian Unit <i>method</i> hitungSkor()	124
Tabel 6.5 Kasus Uji Pengujian Unit <i>method</i> cetakData()	125
Tabel 6.6 Kasus Uji Masuk ke Laman Login	126
Tabel 6.7 Kasus Uji Masuk Sebagai Siswa	126
Tabel 6.8 Kasus Uji Lihat Jadwal Pelajaran	126
Tabel 6.9 Kasus Uji Lihat Info Perubahan Jadwal	127
Tabel 6.10 Kasus Uji Lihat Materi Pelajaran	127
Tabel 6.11 Kasus Uji Lihat Daftar Tuags	127
Tabel 6.12 Kasus Uji Lihat Melakukan Unggah Tugas	127
Tabel 6.13 Kasus Uji Lihat <i>Timeline</i> Akademik	128
Tabel 6.14 Kasus Uji Lihat Daftar Aspirasi	128
Tabel 6.15 Kasus Uji Mengirimkan Pesan Aspirasi	128

Tabel 6.16 Kasus Uji Lihat Profil Akun Pengguna	129
Tabel 6.17 Kasus Uji Melakukan Ubah Profil Akun.....	129
Tabel 6.18 Kasus Uji Melakukan Latihan Soal	129
Tabel 6.19 Kasus Uji Keluar dari Sistem	130
Tabel 6.20 Kasus Uji Masuk ke halaman Login.....	130
Tabel 6.21 Kasus Uji Masuk dalam Sistem Sebagai Pengajar	130
Tabel 6.22 Kasus Uji Lihat Jadwal Pelajaran.....	130
Tabel 6.23 Kasus Uji Melakukan Tambah Info Perubahan Jadwal	131
Tabel 6.24 Kasus Uji Lihat Materi Pelajaran.....	131
Tabel 6.25 Kasus Uji Lihat Daftar Tugas	131
Tabel 6.26 Kasus Uji Menambah Daftar Tugas.....	132
Tabel 6.27 Kasus Uji Melihat Berkas Tugas	132
Tabel 6.28 Kasus Uji Unduh Berkas Tugas	132
Tabel 6.29 Kasus Uji Melihat <i>Timeline</i> Akademik.....	133
Tabel 6.30 Kasus Uji Menambah Kegiatan Akademik	133
Tabel 6.31 Kasus Uji Melihat Daftar Aspirasi	133
Tabel 6.32 Kasus Uji Melakukan Pengiriman Pesan Aspriasi	134
Tabel 6.33 Kasus Uji Melakukan Pendaftaran Akun Siswa	134
Tabel 6.34 Kasus Uji Keluar dari Sistem	134
Tabel 6.35 Hasil Pengujian Validasi Pengguna Siswa	135
Tabel 6.36 Hasil Pengujian Validasi Pengguna Pengajar	142
Tabel 6.37 Daftar <i>Task</i> Pengujian <i>Task Scenario</i> (Siswa).....	150
Tabel 6.38 Daftar <i>Task</i> Pengujian <i>Task Scenario</i> (Pengajar)	151
Tabel 6.39 Hasil Pengujian <i>Task Scenario</i> (Siswa)	153
Tabel 6.40 Hasil Pengujian <i>Task Scenario</i> (Pengajar)	154
Tabel 6.41 Hasil Kuesioner USE	155
Tabel 6.42 Hasil Uji Tes <i>Compatibility</i>	157
Tabel 6.43 Rata-Rata Hasil <i>Task Scenario</i>	158
Tabel 6.44 Presentasi Kelayakan Hasil Kuesioner USE	159
Tabel 6.45 Rata-Rata Hasil Kuesioner USE.....	161

DAFTAR GAMBAR

Gambar 1.1 Infografik pengguna <i>Smartphone</i> di Indonesia	2
Gambar 2.1 Proses <i>User-Centered Design</i>	12
Gambar 3.1 Diagram Metodologi Penelitian	18
Gambar 4.1 Diagram Use Case	31
Gambar 4.2 Diagram Use Case (Iterasi 1)	32
Gambar 4.3 Diagram Use Case (Iterasi 2)	33
Gambar 5.1 Arsitektur Sistem Aplikasi <i>Mobile</i>	46
Gambar 5.2 Arsitektur Sistem Aplikasi E-Learning SMANTIGDA	47
Gambar 5.3 Perancangan Basis Data Akun Siswa	48
Gambar 5.4 Perancangan Basis Data Aspirasi	48
Gambar 5.5 Perancangan Basis Data Info Jadwal	49
Gambar 5.6 Perancangan Basis Data <i>Timeline</i> Akademik	50
Gambar 5.7 Perancangan Basis Data Berkas Tugas.....	51
Gambar 5.8 Perancangan Basis Data Tugas Siswa	52
Gambar 5.9 Perancangan Basis Data ER Diagram Latihan Soal	53
Gambar 5.10 Perancangan <i>Class Diagram</i>	54
Gambar 5.11 Perancangan <i>Class Diagram (Iterasi 1)</i>	55
Gambar 5.12 Perancangan <i>Class Diagram (Iterasi 2)</i>	55
Gambar 5.13 Perancangan <i>Class Diagram Paket KritikSaran</i>	56
Gambar 5.14 Perancangan <i>Class Diagram Paket Aspriasi (Iterasi 1)</i>	56
Gambar 5.15 Perancangan <i>Diagram Class Package JadwalPelajaran</i>	57
Gambar 5.16 Perancangan <i>Class Diagram Paket Jadwal Pelajaran (Iterasi 1)</i>	57
Gambar 5.17 Perancangan <i>Class Diagram Paket LatihanSoal</i>	58
Gambar 5.18 Perancangan <i>Class Diagram Paket Materi Pelajaran</i>	59
Gambar 5.19 Perancangan <i>Class Diagram Paket Profil</i>	60
Gambar 5.20 Perancangan <i>Class Paket Profil (Iterasi 1)</i>	60
Gambar 5.21 Perancangan <i>Class Diagram Paket KalenderAkademik</i>	61
Gambar 5.22 Perancangan <i>Class Diagram Paket TimelineAkademik (Iterasi 1)</i> ...	61
Gambar 5.23 Perancangan <i>Class Diagram Paket Tugas</i>	62
Gambar 5.24 Perancangan <i>Class Paket Tugas (Iterasi 1)</i>	63

Gambar 5.25 Perancangan <i>Class Diagram Paket UserAuth</i>	63
Gambar 5.26 Perancangan <i>Class Diagram Paket UserAuth</i> (Iterasi 1).....	64
Gambar 5.27 Diagram <i>Sequence</i> Ungah Berkas Tugas	65
Gambar 5.28 Diagram <i>Sequence</i> Tambah Daftar Tugas	66
Gambar 5.29 Diagram <i>Sequence</i> Kirim dan Lihat Aspirasi	66
Gambar 5.30 Algoritme Menampilkan Pertanyaan Latihan Soal	67
Gambar 5.31 Algoritme Perhitungan Skor Latihan Soal	67
Gambar 5.32 Algoritme Cetak Data Skor Latihan Soal	68
Gambar 5.33 <i>Screenflow</i> Aplikasi	69
Gambar 5.34 <i>Screenflow</i> Aplikasi (Iterasi 1)	70
Gambar 5.35 <i>Screenflow</i> Aplikasi (Iterasi 2)	71
Gambar 5.36 Rancangan Antarmuka <i>Logon Screen</i>	72
Gambar 5.37 Rancangan Antarmuka <i>Login Screen</i>	73
Gambar 5.38 Rancangan Antarmuka Halaman Utama Siswa	73
Gambar 5.39 Rancangan Antarmuka Halaman Utama Siswa (Iterasi 1)	74
Gambar 5.40 Rancangan Antarmuka Halaman Utama Pengajar	74
Gambar 5.41 Rancangan Antarmuka Halaman Utama Pengajar (Iterasi 1)	75
Gambar 5.42 Rancangan Antarmuka Tugas (Siswa)	75
Gambar 5.43 Rancangan Antarmuka Unggah Tugas	76
Gambar 5.44 Rancangan Antarmuka Jadwal Pelajaran Siswa	76
Gambar 5.45 Rancangan Antarmuka Jadwal Pelajaran Siswa (Iterasi 1)	77
Gambar 5.46 Rancangan Antarmuka Jadwal Pelajaran Pengajar	77
Gambar 5.47 Rancangan Antarmuka Jadwal Pelajaran Pengajar (Iterasi 1).....	78
Gambar 5.48 Rancangan Antarmuka Tambah Info	78
Gambar 5.49 Rancangan Antarmuka Submenu Materi.....	79
Gambar 5.50 Rancangan Antarmuka Materi Pelajaran	79
Gambar 5.51 Rancangan Antarmuka Timeline Akademik Siswa	80
Gambar 5.52 Rancangan Antarmuka Timeline Akademik Siswa (Iterasi 1).....	80
Gambar 5.53 Rancangan Antarmuka Timeline Akademik Pengajar.....	81
Gambar 5.54 Rancangan Antarmuka Timeline Akademik Pengajar(Iterasi 1).....	81
Gambar 5.55 Rancangan Antarmuka Tambah Kegiatan.....	82
Gambar 5.56 Rancangan Antarmuka Akun.....	82

Gambar 5.57 Rancangan Antarmuka Ubah Profil	83
Gambar 5.58 Rancangan Antarmuka Aspirasi.....	83
Gambar 5.59 Rancangan Antarmuka Aspirasi (Iterasi 1)	84
Gambar 5.60 Rancangan Antarmuka Tugas (Pengajar)	85
Gambar 5.61 Rancangan Antarmuka Lihat Berkas.....	85
Gambar 5.62 Rancangan Antarmuka Pendaftaran Akun.....	86
Gambar 5.63 Rancangan Antarmuka Pendaftaran Akun (Iterasi 1)	86
Gambar 5.64 Rancangan Antarmuka Pendaftaran.....	87
Gambar 5.65 Rancangan Antarmuka Menu Latihan Soal.....	87
Gambar 5.66 Rancangan Antarmuka Latihan Soal	88
Gambar 5.67 Rancangan Antarmuka Hasil Latihan Soal.....	88
Gambar 5.68 Implementasi Basis Data Akun Siswa	90
Gambar 5.69 Implementasi Basis Data Aspirasi.....	91
Gambar 5.70 Implementasi Basis Data Tugas Siswa	91
Gambar 5.71 Implementasi Basis Data <i>Uploads</i>	91
Gambar 5.72 Implementasi Basis Data Info Jadwal	92
Gambar 5.73 Implementasi Basis Data Timeline Akademik	92
Gambar 5.74 Implementasi Struktur Basis Data Latihan Soal	92
Gambar 5.75 Implementasi Basis Data Latihan Soal	93
Gambar 5.76 Implementasi <i>Method</i> tambahTugas()	96
Gambar 5.77 Implementasi <i>Method</i> uploadFile()	98
Gambar 5.78 Implementasi <i>Method</i> addAspirasi()	99
Gambar 5.79 <i>Screenflow</i> Antarmuka.....	102
Gambar 5.80 Implementasi Antarmuka <i>Logon Screen</i>	103
Gambar 5.81 Implementasi Antarmuka Login Siswa.....	104
Gambar 5.82 Implementasi Antarmuka Login Pengajar.....	104
Gambar 5.83 Implementasi Antarmuka Halaman Utama Siswa.....	104
Gambar 5.84 Implementasi Antarmuka Halaman Utama Pengajar.....	104
Gambar 5.85 Implementasi Antarmuka Halaman Jadwal Pelajaran Siswa	105
Gambar 5.86 Implementasi Antarmuka Halaman Jadwal Pelajaran Siswa 2	105
Gambar 5.87 Implementasi Antarmuka Halaman Jadwal Pelajaran Pengajar 2	105
Gambar 5.88 Implementasi Antarmuka Halaman Jadwal Pelajaran Pengajar ...	105

Gambar 5.89 Implementasi Antarmuka Tambah Info Perubahan Jadwal	106
Gambar 5.90 Implementasi Antarmuka Halaman Mata Pelajaran 2	107
Gambar 5.91 Implementasi Antarmuka Halaman Mata Pelajaran	107
Gambar 5.92 Implementasi Antarmuka Halaman Pengajar 2	107
Gambar 5.93 Implementasi Antarmuka Halaman Pengajar	107
Gambar 5.94 Implementasi Antarmuka Halaman Tugas Pengajar (notifikasi)...	107
Gambar 5.95 Implementasi Antarmuka Lihat Berkas.....	108
Gambar 5.96 Implementasi Antarmuka Tugas Siswa	108
Gambar 5.97 Implementasi Antarmuka Halaman Unggah Tugas	109
Gambar 5.98 Implementasi Antarmuka Halaman Unggah Tugas 2	109
Gambar 5.99 Implementasi Antarmuka Timeline Akademik (Siswa).....	110
Gambar 5.100 Implementasi Antarmuka Timeline Akademik (Pengajar)	110
Gambar 5.101 Implementasi Antarmuka Tambah Kegiatan Akademik	110
Gambar 5.102 Implementasi Antarmuka Aspirasi	111
Gambar 5.103 Implementasi Antarmuka Akun.....	112
Gambar 5.104 Implementasi Antarmuka Ubah Akun	112
Gambar 5.105 Implementasi Antarmuka Latihan Soal 1	113
Gambar 5.106 Implementasi Antarmuka Latihan Soal 2	113
Gambar 5.107 Implementasi Antarmuka Latihan Soal 3	113
Gambar 5.108 Implementasi Antarmuka Latihan Soal 4	113
Gambar 5.109 Implementasi Antarmuka Hasil Latihan Soal	114
Gambar 5.110 Implementasi Antarmuka Pendaftaran Akun	115
Gambar 5.111 Implementasi Antarmuka Pendaftaran	115
Gambar 6.1 Kode Program <i>method</i> <i>getPDF()</i>	117
Gambar 6.2 <i>Flowgraph</i> <i>method</i> <i>getPDF()</i>	117
Gambar 6.3 Kode Program <i>method</i> <i>uploadFile()</i>	119
Gambar 6.4 <i>Flowgraph</i> <i>method</i> <i>uploadFile()</i>	120
Gambar 6.5 Kode Program <i>method</i> <i>updatePertanyaan</i>	121
Gambar 6.6 <i>Flowgraph</i> <i>method</i> <i>updatePertanyaan()</i>	121
Gambar 6.7 Kode Program <i>method</i> <i>hitungSkor()</i>	123
Gambar 6.8 <i>Flowgraph</i> <i>method</i> <i>hitungSkor()</i>	123
Gambar 6.9 Kode Program <i>method</i> <i>cetakData</i>	124

Gambar 6.10 *Flowgraph method cetakData()* 125



DAFTAR LAMPIRAN

LAMPIRAN A PERSONA CALON PENGGUNA APLIKASI.....	167
A.1 Persona Siswa SMAN 3 Sidoarjo.....	167
A.2 Persona Pengajar SMAN 3 Sidoarjo	168
LAMPIRAN B FORM KUESIONER USE	170
LAMPIRAN C DAFTAR PERTANYAAN WAWANCARA	172



BAB 1 PENDAHULUAN

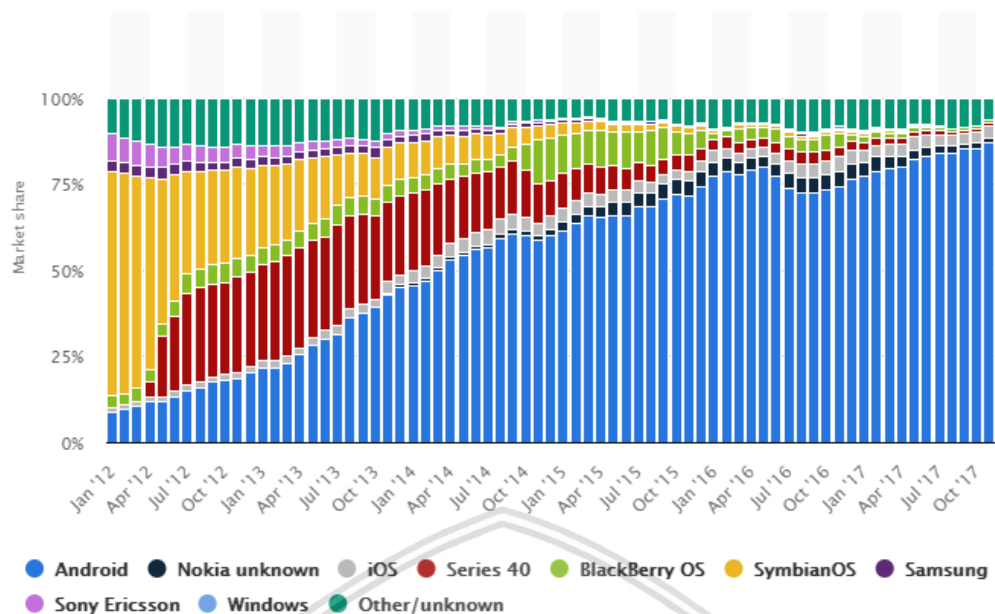
1.1 Latar belakang

Pendidikan merupakan kebutuhan yang harus dipenuhi oleh setiap manusia untuk menjalani kehidupan mereka, dengan pendidikan manusia dapat berkembang sesuai dengan pandangan hidup mereka. Menurut *Dictionary of Education* menjelaskan bahwa pendidikan merupakan sebuah proses seseorang untuk mengembangkan kemampuan sikap mereka, serta tingkah laku mereka dalam hidup bermasyarakat. Teknologi juga berperan penting dalam pendidikan, teknologi dapat meningkatkan cara belajar siswa untuk mendapatkan hasil yang lebih maksimal (Khalkhali, 2014).

Perkembangan dunia internet memiliki dampak pada sistem pembelajaran, dengan adanya dampak seperti itu maka terciptalah model pembelajaran baru yang disebut *e-learning*. Model pembelajaran ini mengarahkan sistem pembelajaran yang tidak menggunakan metode yang tradisional lagi, namun merujuk pada penyampaian materi pembelajaran dengan menggunakan media elektronik, dan *e-learning* juga merupakan salah satu model pembelajaran yang sangat populer karena keuntungan dari model pembelajaran ini partisipan dapat mengakses program dimanapun kapanpun, tidak seperti model pembelajaran tradisional yang terpaku pada waktu dan lokasi (Khalkhali, 2014).

SMAN 3 Sidoarjo merupakan salah satu sekolah negeri di Kota Sidoarjo yang memiliki beberapa fasilitas dalam bidang edukasi salah satunya adalah *e-learning*. Sistem *e-learning* yang sudah berjalan beberapa tahun belakangan ini hanya dipergunakan untuk kepentingan ulangan harian, UTS dan UAS saja. Lima siswa yang dijadikan perona dalam penelitian ini, mereka terkadang juga merasa terbebani jika tim pengajar memberikan ulangan harian secara tiba-tiba disaat kondisi siswa tersebut tidak sedang dalam keadaan siap atau jauh dari PC atau Laptop. Hal itulah yang terkadang sangat memberatkan siswa dalam hal mobilitas yang mengharuskan siswa tetap siaga di depan PC atau laptop dan terkoneksi dengan internet secara berkala. Seringnya siswa melewatkan tugas juga merupakan suatu masalah karena kurangnya pemberitahuan secara *real-time* karena siswa juga kerap kali harus melakukan pengecekan secara berkala untuk mengakses *e-learning* agar mengetahui pembaharuan tugas dari tim pengajar. Untuk itu siswa SMAN 3 Sidoarjo memerlukan sebuah sarana untuk membantu siswa agar tidak tertinggal untuk memantau pembaharuan di *e-learning*.

Pada tahun 2017 ini perangkat *mobile* bukanlah hal yang tidak biasa lagi. Berdasarkan presentase yang diperoleh dari situs web Statista: The Statistic Portal menunjukkan bahwa lebih dari 75% pengguna *smartphone* di Indonesia menggunakan *smartphone* ber-sistem operasi Android dibanding dengan *smartphone* dengan sistem operasi lainnya (Statista, 2017).



Gambar 1.1 Infografik pengguna *Smartphone* di Indonesia

Sumber: (Statista, 2017).

Dengan data yang ada di lapangan, dari tahun-ke-tahun yang semakin bertambah pengguna *smartphone* android, secara tidak langsung sudah pasti banyak aplikasi yang disediakan untuk *smartphone* Android, tidak terkecuali pada aplikasi dengan kategori pendidikan, namun masih banyak pengguna yang terkadang masih susah memahami dan menggunakan aplikasi yang ada. Maka *smartphone* dapat menjadi solusi untuk siswa agar tetap bisa mendapatkan pembaharuan informasi dari *e-learning*.

Dengan kondisi *e-learning* sekolah saat ini yang tidak dipergunakan dan dikembangkan dengan maksimal oleh pihak sekolah, pada penelitian ini memiliki harapan agar aplikasi *e-learning* terlihat lebih *user-friendly*, memiliki tingkat *usability* yang tinggi serta memberikan kesesuaian aplikasi dengan kebutuhan yang dibutuhkan oleh pengguna dengan menerapkan pendekatan *User Centered Design* pada pembangunan aplikasinya. UCD merupakan salah satu metode pendekatan dengan konsep pengguna yang dioptimalkan untuk *end-user* serta ditekankan pada bagaimana kebutuhan atau keinginan pada setiap *end-user*, dan dirancang sesuai *behavior end-user* agar pengguna tidak memaksa untuk mengubah perilakunya saat menggunakan produk yang akan dibangun. Dari tujuan pembangunan hingga penentuan desain aplikasi tergantung dari *experience* pengguna. UCD juga merupakan suatu proses interaktif dimana langkah perancangan dan evaluasi dibuat dalam permulaan proyek hingga proses implementasi (Devi, et al., 2012). Selain akan menguji tingkat *usability* dari aplikasi yang akan kembangkan, akan dilakukan juga pengujian *compatibility* yang penting untuk menguji apakah aplikasi dapat berjalan baik di berbagai *device* dikarenakan banyaknya *device* siswa dan pengajar yang memiliki bermacam-macam spesifikasinya seperti versi OS Android, ukuran layar *device* dan resolusi layar-nya (Pusuluri, 2006).

Dari latar belakang yang sudah dipaparkan, penulis melakukan penelitian untuk membangun aplikasi pada *platform* Android dengan desain antarmuka yang menerapkan pendekatan *User Centered Design* untuk memberikan kemudahan dalam proses pembelajaran untuk siswa dan pengajar.

1.2 Rumusan masalah

Berdasarkan latar belakang diatas maka rumusan masalahnya dapat disimpulkan sebagai berikut:

1. Bagaimana analisis, perancangan, implementasi dan pengujian pada pembangunan aplikasi *e-learning* yang dapat dijalankan pada perangkat Android dengan menerapkan pendekatan *User Centered Design*?
2. Bagaimana tingkat kemudahan pengguna dalam menggunakan sistem *e-learning* pada perangkat Android?

1.3 Tujuan

Berdasarkan rumusan masalah diatas, maka dapat tujuannya dapat disimpulkan sebagai berikut:

1. Melakukan analisis, perancangan, implementasi dan pengujian pada pembangunan aplikasi *e-learning* yang dapat dijalankan pada perangkat Android dengan menerapkan pendekatan *User Centered Design*.
2. Mengetahui tingkat kemudahan pengguna dalam menggunakan sistem *e-learning* pada perangkat Android.

1.4 Manfaat

Manfaat dari penelitian ini diantaranya dapat disimpulkan sebagai berikut:

1. Siswa dan pengajar dapat terbantu dalam kegiatan pembelajaran melalui perangkat Android.
2. Mempermudah siswa dan pegajar untuk saling memberikan informasi terkait kegiatan pembelajaran serta penyampaian aspirasi untuk memajukan sekolah.

1.5 Batasan masalah

Agar permasalahan yang telah dirumuskan lebih terfokus dan tidak terjadi pelebaran topik pembahasan, maka penelitian ini memiliki batasan masalah sebagai berikut:

1. Perancangan serta implementasi aplikasi *e-learning* ini hanya digunakan pada *mobile device* menggunakan Bahasa Java pada sistem operasi Android.
2. Studi kasus penelitian ditujukan kepada siswa dan pengajar SMAN 3 Sidoarjo.
3. Aplikasi ini menggunakan Firebase sebagai *database* penyimpanan data.

4. Pengumpulan berkas tugas yang akan dikumpulkan sudah ada di dalam *internal* atau *extenal storage device* pengguna sebelumnya.

1.6 Sistematika pembahasan

Berikut merupakan sistematika penulisan yang akan disusun untuk menyelesaikan penelitian ini, yaitu:

BAB 1 PENDAHULUAN

Bab ini berisi tentang latar belakang, rumusan masalah, tujuan, manfaat, batasan masalah dan sistematika pembahasan.

BAB 2 LANDASAN KEPUSTAKAAN

Bab ini membahas tentang pengertian dan konsep teori yang digunakan sebagai dasar untuk menunjang penyusunan tugas akhir.

BAB 3 METODOLOGI

Membahas metode yang digunakan dalam melakukan penelitian tugas akhir mulai dari tahapan studi literatur, analisis kebutuhan, perancangan, implementasi, hingga pengujian aplikasi yang akan dibuat.

BAB 4 REKAYASA KEBUTUHAN

Menguraikan tentang analisis kebutuhan sistem dari Aplikasi *e-learning* dengan pendekatan *User Centered Design* yang terdiri atas identifikasi aktor, gambaran umum aplikasi, analisis kebutuhan fungsional dan non-fungsional.

BAB 5 PERANCANGAN DAN IMPLEMENTASI

Bab ini membahas tentang perancangan sistem dan implementasi perangkat lunak sesuai dengan perancangan sistem yang telah dibuat.

BAB 6 PENGUJIAN

Membahas tentang kesimpulan dan saran sesuai hasil penelitian berdasarkan uraian-uraian dari bab-bab sebelumnya untuk pengembangan lebih lanjut.

BAB 7 PENUTUP

Bab ini membahas tentang kesimpulan dan saran sesuai hasil penelitian berdasarkan uraian-uraian dari bab-bab sebelumnya untuk pengembangan lebih lanjut.

BAB 2 LANDASAN KEPUSTAKAAN

Pada bab ini, penulis membahas tinjauan pustaka berupa kajian pustaka yang membahas tentang penelitian terdahulu yang sudah pernah ada serta dasar teori yang digunakan untuk menyusun dan pelaksanaan penelitian yang diusulkan. Kajian pustaka yang dijadikan sebagai acuan adalah penelitian yang berhubungan *e-learning*. Untuk dasar teori yang digunakan untuk penunjang penelitian adalah konsep dasar dari *e-learning* itu sendiri.

2.1 Kajian Pustaka

Kajian pustaka pada penelitian ini mencakup tentang penelitian sebelumnya yang berkaitan dengan *e-learning* dengan menerapkan basis *learning management system* pada aplikasi perangkat bergerak. Penelitian terdahulu Rancang Bangun *E-Learning* Berbasis Moodle menggunakan Aplikasi *Mobile* Android. Pada peneliti sebelumnya berfokus pada penggunaan *Learning Management System* Moodle yang digunakan (Putra, 2013). Pada penelitian tersebut penulis mencoba untuk mengembangkan aplikasi *e-learning* menggunakan *learning management system* dari Moodle yang diimplementasikan pada perangkat bergerak dengan menerapkan semua fitur dari *Lms* Moodle dalam pengembangannya, pada penelitian ini kebutuhan pada aplikasi dapat tidak sesuai dengan pengguna yang menggunakan aplikasinya, karena semua fitur pada *learning management system* Moodle diterapkan pada penelitian ini.

Peneliti terdahulu yang kedua yaitu pada Rancang Bangun *Push Notification* Berbasis Moodle pada perangkat Android ini berfokus pada notifikasi secara *realtime* pada aplikasi perangkat berbergerak pengguna untuk menerima informasi-informasi terkini tentang pembelajaran (Pramita, 2015). Pengembangan aplikasi tersebut hampir sama dengan penelitian pertama, namun pada penelitian ini terdapat tambahan fitur *notifikasi real-time* pada aplikasi yang dikembangkan, dengan menerapkan semua fitur yang terdapat pada *learning management system* Moodle, aplikasi yang digunakan pada pengguna dapat tidak sesuai dengan kebutuhan yang dibutuhkan oleh penggunanya nanti karena pada penelitian ini mengimplementasikan semua fitur yang terdapat pada *learning management system* Moodle.

Berbeda dengan sistem yang dibuat peneliti yang fokus pada bagaimana cara penyampaian fitur *e-learning* dengan menggunakan metode UCD serta memberikan kemudahan bagi pengguna dalam kegiatan pembelajaran sekolah baik dari pengajar maupun siswa itu sendiri, selain itu terdapat juga notifikasi tugas dan fitur unggah tugas untuk memudahkan proses pembelajaran diluar sekolah.

2.2 E-Learning

Model pembelajaran yang populer saat ini adalah model pembelajaran *e-learning*, yang sistem pembelajarannya tidak menggunakan sistem pembelajaran yang tradisional lagi, melainkan sudah merujuk pada model pembelajaran yang memanfaatkan media elektronik sebagai penunjang. *E-learning* merupakan sebuah istilah pembelajaran yang telah didukung oleh teknologi sebagai medianya, seperti telepon, *audio*, *video teleconference*, transmisi satelit dan *web-based training* dengan bantuan komputer sebagai alat pemrosesnya (Soekartawi, 2002).

Dari sumber lain menjelaskan *e-learning* merupakan suatu metode pengajaran atau pembelajaran dengan menggunakan rangkaian elektronik (LAN, WAN atau internet) untuk penyampaian materi pembelajarannya (Jaya Kumar C, 2002). Selain itu ada juga yang menyebutkan bahwa *e-learning* ini merujuk pada pemanfaatan teknologi internet untuk memecahkan suatu permasalahan secara menyeluruh untuk meningkatkan pengetahuan dan kinerja (Rosenberg, 2001). Dari berbagai sumber yang ada, maka dapat disimpulkan bahwa *e-learning* merupakan sebuah model pembelajaran dengan memanfaatkan media komputer dan jaringan internet untuk menyampaikan materi atau konten pembelajarannya.

Pada penelitian yang dikerjakan akan mengadaptasi bagaimana model pembelajaran dari *e-learning* itu sendiri dan menerapkannya pada aplikasi yang dibangun serta dapat dijalankan pada perangkat Android.

2.2.1 Konsep E-Learning

Kurangnya metode pengajaran tradisional dikenal kurang efektif jika dibandingkan dengan metode pembelajaran modern. Berbagai hal yang terdapat pada *e-learning* sebagai berikut (Ramdhan, 2011):

1. Soal-soal

Materi yang disediakan dalam bentuk modul, disamping itu juga disediakan soal-soal latihan yang ditampilkan. Hasil dari pengerjaan soal-soal tersebut dapat dijadikan tolak ukur kemampuan siswa.

2. Komunitas

Siswa dapat mengembangkan komunitas sendiri secara *online* sebagai sarana pendukung diskusi dan sarana berbagi informasi.

3. Pengajar *online*

Pemberian arahan untuk para siswa dapat dilakukan secara *online*, menjawab pertanyaan siswa dan membantu para siswa untuk berdiskusi.

4. Kesempatan bekerja sama

Dengan memanfaatkan teknologi dapat bekerja sama belajar dengan pengguna lain tanpa ada kenjalar jarak.

5. Multimedia

Penggunaan media *audio* dan *video* dalam menyampaikan materi dapat menarik minat para siswa untuk melakukan kegiatan pembelajaran

Dalam model pembelajaran *e-learning* memiliki fokus utama, yaitu siswa itu sendiri. Siswa dituntut untuk mandiri serta lebih aktif dalam kegiatan pembelajaran.

2.2.2 Komponen E-Learning

Ada beberapa komponen pendukung untuk membentuk e-learning, antara lain (Ramdhan, 2011):

1. Infrastruktur *e-learning*

Infrastruktur *e-learning* dapat berupa jaringan komputer, pc, internet, pelengkap *multimedia* hingga perangkat bergerak (*smartphone*).

2. Sistem dan Aplikasi *e-learning*

Perangkat lunak dapat membantu membuat bentuk *virtual* dari proses belajar secara konvensional. Dalam manajemen kelas, pembuatan konten berupa materi, forum diskusi, sistem penilaian, ujian *online* dan segala fitur tambahan dapat dihubungkan dalam proses pembelajaran. Banyak sistem perangkat lunak pembelajaran yang telah populer saat ini, seperti Moodle, Dokeos, Atutor, dll. Sistem perangkat lunak pembelajaran tersebut dapat disebut dengan *Learning Management System* (LMS).

3. Konten *e-learning*

Konten pembelajaran *e-learning* berbentuk *multimedia-based content* dan *text-based content*. Kedua bentuk konten tersebut dapat disimpan dalam *Learning Management System* sehingga dapat diakses oleh siswa kapanpun dan dimanapun.

Untuk pelaku atau aktor dari pengguna *e-learning* adalah pengajar yang berperan sebagai pembimbing, siswa yang menerima bahan ajar dari tim pengajar serta *administrator* yang mengelola administrasi dalam kegiatan ajar mengajar.

2.3 Android

Android merupakan sebuah perangkat lunak yang mencakup sistem operasi, *middleware* dan aplikasi kunci. Android dapat dikatakan sebagai sistem operasi berbasis Linux untuk perangkat bergerak (Sunaryo, et al., 2011). Sistem operasi ini memiliki bersifat *open source* dan tidak sulit untuk memahami arsitektur sistem operasi ini, hal tersebut yang menjadikan android sangat cepat berkembang sampai saat ini (Ardianto, et al., 2012). Produk yang dicanangkan oleh Google ini sudah digunakan lebih dari 190 negara di dunia dengan Nougat sebagai generasi ke-7 dari sistem operasi ini (Android, 2016).

Ada beberapa kelebihan Android yang membuat sistem operasi ini berkembang sangat cepat, yaitu (Hamka, et al., 2015):

1. Bersifat *open source*, siapapun dapat mengembangkan sistem operasi ini dengan bebas, tidak perlu membayar royalti berapapun kepada pihak android.
2. Fitur yang disediakan cukup lengkap dan beragam, contohnya *Global Positioning System* (GPS), penyimpanan data secara lokal dengan SQL, dan masih banyak lagi.
3. Grafik pada sistem operasi ini cukup baik, sudah memiliki *antialiased* 2D dan 3D *Accelerated* OpenGL, serta untuk *audio* juga sudah dilengkapi dengan codec untuk dapat menjalankan AVC, MP3, AAC, OGG dan masih banyak lagi.

Selain perangkat bergerak (*smartphone*) juga terdapat beberapa perangkat lainnya yang menggunakan sistem operasi Android seperti televisi, kamera, dan sebagainya.

2.3.1 Jenis Android

Android merupakan salah satu sistem operasi untuk *smartphone* berbasis Linux. Android besutan Google ini ditujukan kepada para pengembang yang ingin menciptakan aplikasi mereka sendiri, karena sistem operasi Android bersifat *open source* (Triadi, 2013).

1. Astro

Android versi paling pertama ini rilis pada 23 september 2008. Pertama kali OS ini dibekali pada *smartphone* HTC Dream.

2. Bender

Versi kedua ini masih versi milestone dari varian Android itu sendiri yang rilis pada 9 Maret 2009.

3. Cupcake

Android varian ini mulai merambah menjadi nama varian pencuci mulut, rilis pada 30 April 2009. Android versi ini memiliki code 1.5.

4. Donut

Varian Android ini mulai rilis pada September 2009 dengan code 1.6.

5. Éclair

Androd versi ini dirilis pada tanggal 9 September 2010. Versi android ini melakukan perbaikan pada pengoptimalan hardware, browser baru, dan masih banyak lagi. Android versi ini memiliki kode rilis 2.0 & 2.1.

6. Froyo

Singkatan dari Frozen Yoghurt, versi android ini rilis pada tanggal 20 Mei 2010 dengan code 2.2.

7. Gingerbread

Android versi ini dirilis pada tanggal 6 Desember 2010, memiliki kode versi 2.3.

8. Honeycomb

Versi ini dirilis khusus untuk pengguna tablet, resmi diluncurkan pada 22 Februari 2011. Android versi ini memiliki kode rilis 3.0.

9. Ice Cream Sandwich

Versi tersebut dirilis pada 19 Oktober 2011. Ice Cream Sandwich ini digunakan sebagai nama alias dari Android versi 4.0.

10. Jelly Bean

Pada Android versi ini Google meningkatkan fungsi input keyboard, melakukan perubahan design UI dan mengoptimalkan Voice Search. Android versi ini memiliki kode versi 4.1.

11. KitKat

Android versi yang rilis pada tanggal 31 Oktober 2013 ini digunakan sebagai nama alias dari sistem Android versi 4.4.x.

12. Lollipop

Android versi 5.0 ini dirilis pada tanggal 15 Oktober 2014.

13. Marshmallow

Android versi 6.0 ini rilis pada tanggal 19 Agustus 2015.

14. Nougat

Android versi 7.0 ini rilis pada tanggal 22 Agustus 2016 dengan membawa perubahan UI yang lebih sederhana dan bersih.

15. Oreo

Android versi 8.0 ini merupakan android versi terbaru dengan membawa perubahan pada daya yang digunakan oleh sistem. Android versi ini rilis pada 21 Agustus 2017.

2.4 Firebase

Firebase merupakan sebuah layanan dari Google yang menyediakan layanan basis data secara *realtime*. Basis data ini disimpan dalam format JSON serta melakukan sinkronisasi secara berkala untuk selalu mendapat keterhubungan dengan klien (KhedKar & Thube, 2017). Firebase merupakan layanan basis data yang dapat digunakan dari berbagai macam platform, antara lain Android, iOS, serta aplikasi Web. Saat aplikasi *cross-platform* dibuat dengan SDK Android, iOS dan JavaScript, semua klien akan berbagi sebuah *Instance Realtime Database* dan akan menerima update secara otomatis.

Saat API diimplementasikan pada aplikasi *cross-platform*, Firebase dapat memberikan fitur dengan bagis kode sederhana. Fitur-fitur yang ditawarkan Firebase yaitu Autentifikasi, *Hosting*, *Messaging*, *Analytics*, *Storage*, *Real-time Database*, *Crash Reporting*, *App Indexing* dan *Admob*. Masing-masing dari fitur tersebut dapat dijabarkan sebagai berikut:

1. Authentication

Authentication ini merupakan salah satu layanan yang diberikan oleh Firebase untuk memberikan izin hanya kepada pengguna tertentu yang sudah diberikan izin untuk mengakses aplikasi. Autentikasi ini menyediakan fitur *login* dan *register* melalui *email*, nomor ponsel, anonim hingga mendukung beberapa media sosial seperti Gmail, Github, Twitter, Facebook untuk mempermudah pengguna menggunakan aplikasi.

2. Hosting

Hosting merupakan konten web kelas produksi yang ditujukan pada pengembang aplikasi, layanan ini dapat dengan cepat dan efektif mengirim aplikasi web dan konten statis ke *Content Delivery Network* (CDN) dengan satu perintah. *Hosting* berisi dukungan domain khusus, Sertifikat SSL *Global CDN* dan *Auto Provisioned*.

3. Messaging

Messaging adalah salah satu layanan yang disediakan oleh Firebase untuk mengirim pesan lintas platform tanpa menggunakan biaya. Dengan Firebase Cloud Messaging pengembang dapat melakukan aktivitas berkirim pesan pada pengguna.

4. Analytics

Analytics merupakan fitur yang disediakan oleh Firebase untuk membantu pengembangan mengetahui seberapa jauh aplikasinya mendapatkan interaksi dari pengguna.

5. Storage

Storage merupakan fitur yang disediakan oleh Firebase yang berfungsi sebagai media penyimpanan basis data secara *real-time*. *Storage* ini dapat menyimpan berbagai macam berkas seperti gambar, audio, video serta berkas lainnya serta memungkinkan pengguna aplikasi dapat melakukan aktifitas unggah dan unduh secara langsung pada layanan ini.

6. Real-time Database

Firebase real-time Database adalah database yang diposisikan pada cloud. Datanya disimpan dalam bentuk JSON dan disinkronkan secara *real-time* tiap kali pengguna aplikasi terhubung. Ketika pengembang membuat aplikasi lintas platform dengan SDK Android, iOS, dan

JavaScript, semua klien akan berbagi sebuah *instance Realtime Database* dan menerima *update* data terbaru secara otomatis.

7. Crash Reporting

Firebase Crash Reporting membuat detail laporan tentang terjadinya *error* pada aplikasi yang dibuat oleh pengembang. *Error* yang terjadi akan dikelompokkan berdasarkan *error-error* yang serupa atau sejenis serta diurutkan berdasarkan tingkat terparah terjadinya *error* pada aplikasi. Selain otomatis, pengembang juga dapat membuat *log* untuk mempermudah menemukan masalah yang menyebabkan aplikasi tidak bekerja.

8. App Indexing

App Indexing sangat berguna bagi pengembang untuk memasukkan aplikasi yang telah dikembangkan dalam penelusuran Google. Layanan ini juga menawarkan pelengkap otomatis kueri untuk membantu pengguna menemukan hal yang diinginkan lebih cepat.

9. AdMob

Layanan AdMob ini digunakan untuk mendapatkan pendapatan dari aplikasi yang telah dibuat oleh pengembang. Layanan ini dapat ditambah kemampuannya untuk lebih analitis pada data pengguna aplikasi jika menggunakan layanan ini dengan Google Analytics.

2.5 OneSignal

OneSignal merupakan sebuah *service push notification* untuk *cross-platform* seperti *website* dan aplikasi *mobile*. OneSignal telah *support* dengan berbagai *native* dan *mobile platform* yang menyediakan RESTful server API, *online dashboard* untuk melihat performa, statistik pengguna dan operasi *push notifikasi* langsung dari *web service*-nya (OneSignal, 2018). Terdapat beberapa kelebihan yang dimiliki oleh OneSignal, antara lain: (Gumelar, 2017)

1. Sudah mendukung implemetasi *interface* melalui GCM/FCM milik Google serta APNS milik Apple.
2. Menyediakan *tools* tambahan untuk marketing seperti *A/B testing*, *segment targeting*, *variable-substitution*, *localization*, *drop marketing* dan konversi *tracking*.
3. OneSignal menyediakan SDK sebagian besar *cross-platform mobile development* seperti Unity, PhoneGap, Cordova, Ionic, ReactNative, Intel XDK, Corona, Xamarin, Marmalade, Adobe Air dan Web Push.

2.6 SQLite

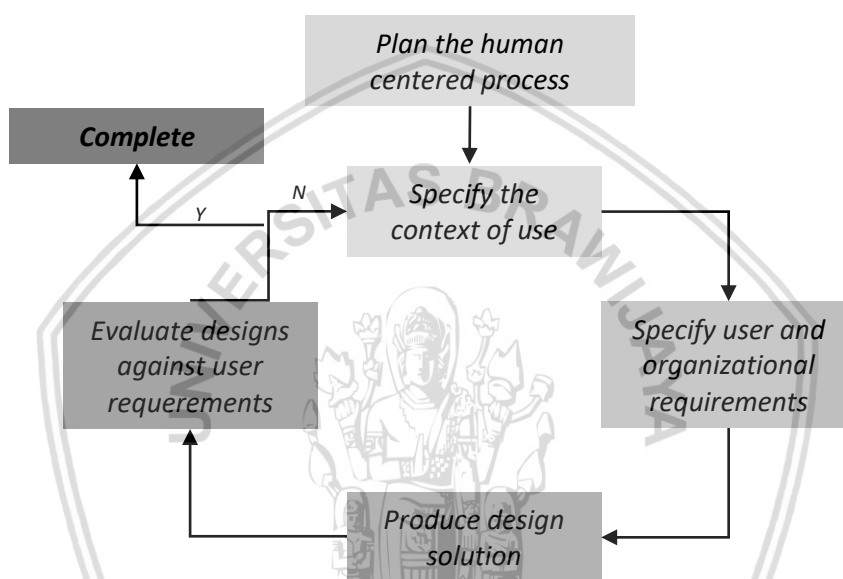
SQLite adalah sebuah Relational Database Management Server (RDBMS) pada sistem operasi Android. SQLite dapat ditemukan pada android Runtime, jadi setiap aplikasi android dapat membuat basis data SQLite (ST, 2015). Ada

beberapa kelebihan SQLite jika digunakan dalam pengembangan aplikasi android, antara lain:

- Penggunaan tanpa perlu melakukan konfigurasi ulang.
- Bersifat *Open Source*.
- Menyediakan fungsionalitas database dalam satu set *library*.

2.7 User Centered Design (UCD)

Filosofi perancangan dengan menempatkan pengguna sebagai pusat dari proses pengembangan sistem.



Gambar 2.1 Proses User-Centered Design

Sumber : (ISO 13407:1999)

Dengan gambar diatas menjelaskan tentang proses UCD berdasarkan ISO 13407:1999.

1. Specify the Context of Use

Tahap ini masuk pada tahap analisis data pada metodologi yang dilakukan dengan memahami dan menentukan konteks yang diinginkan oleh pengguna serta identifikasi siapa saja pengguna aplikasinya, memahami karakteristik pengguna, apa yang dilakukan pengguna, apa yang disukai pengguna, memahami lingkungan yang tepat untuk menggunakan sistem yang akan dibuat. Hal ini sangat penting untuk menentukan kebutuhan sistem dengan optimal.

2. Specify User and Organizational Requirements

Pada tahap ini juga masuk pada tahap analisis data yang membuat pernyataan eksplisit kebutuhan pengguna terkait kualitas perancangan interaksi manusianya. Pada tahap ini dilakukan identifikasi kebutuhan

dari pengguna. Peneliti melakukan dengan wawancara untuk mendapatkan informasi tentang kebutuhan fungsional dan non-fungsional yang akan diterapkan pada aplikasi.

3. *Produce Design Solution*

Tahap ini masuk pada tahap perancangan pada metodologi penelitian, dengan menggunakan dasar pengetahuan yang ada dapat membantu mengembangkan solusi perancangan yang lebih konkrit dengan cara membuat simulasi, prototype dan sebagainya. Kemudian menunjukkan prototype tersebut kepada pengguna dan mengamatinnya, serta menerima umpan balik dari pengguna. Proses ini akan diulang berulang kali hingga tujuan dari perancangan terpenuhi (Preece, 1994).

Pada tahap ini peneliti membangun desain aplikasi yang akan diimplementasikan sesuai kebutuhan fungsional dan non-fungsionalnya.

4. *Evaluate Designs Against User Requirements*

Tahap ini juga masuk pada tahap perancangan untuk melakukan evaluasi sebelum masuk pada tahap implementasi, tahap ini digunakan untuk mengetahui apakah kebutuhan dari pengguna telah tercapai. Untuk mengetahui apakah evaluasi ini telah tercapai atau tidak, tahap evaluasi ini peneliti menggunakan kuisioner untuk diberikan kepada calon pengguna.

2.7.2 Prinsip *User Centered Design*

UCD juga memiliki beberapa prinsip yang harus diperhatikan, antara lain (Widhiarso, et al., 2007):

1. Fokus pada pengguna

Untuk melakukan perancangan, yang dilakukan harus berhubungan langsung dengan penggunaannya melalui beberapa cara yaitu wawancara, *survey*, hingga melakukan *workshop* perancangan untuk memahami karakteristik dari penggunaannya lebih jelas. Pada prinsip ini dilakukan pengambilan data, menganalisisnya dan mengintegrasikannya ke dalam perancangan.

2. Perancangan terintegrasi

Perancangan dilakukan harus mencakup antarmuka pengguna, bantuan terhadap sistem yang telah dibuat, serta prosedur instalasi dan konfigurasi.

3. Pengujian pengguna dilakukan secara berkelanjutan

Wawasan akan siapa pengguna sistem akan sangat membantu untuk memecahkan permasalahan yang ada

4. Perancangan interaktif

Melakukan pendefinisian, perancangan dan diuji berulang kali. Hasil uji yang dilakukan akan dijadikan sebagai bahan pengembangan sistem.

2.7.3 Target *User Centered Design*

User Centered Design (UCD) memiliki target pengguna berasal dari kerabat terdekat dari sasaran pengguna aplikasi yang diteliti, tentunya disesuaikan dengan kebutuhannya. Pendekatan UCD ini dioptimalkan untuk *end-user*, memahami apa yang *user* butuhkan dan inginkan terhadap suatu aplikasi yang diteliti. Dirancang sesuai *behavior* pengguna dan tidak memaksa untuk merubah *behavior* pengguna untuk menggunakan aplikasi yang sedang diteliti. (Barnum, 2011).

2.8 Pengujian Perangkat Lunak

Pengujian perangkat lunak bertujuan untuk mengevaluasi kemampuan aplikasi. Selain mengevaluasi juga menguji aplikasi untuk menemukan kesalahan-kesalahan pada aplikasi yang dibuat. Pengujian sangat penting karena dapat meningkatkan kualitas perangkat lunak.

2.8.1 Pengujian Unit

Pengujian unit atau pengujian modul ini merupakan pengujian subprogram atau prosedur individual dari sebuah program. Pengujian ini terdapat beberapa motivasi yaitu salah satunya mempermudah tugas *debugging* dalam mencari kesalahan pada kode program untuk membantu memenuhi persyaratan penggunanya (Mayers, 2004).

2.8.2 Pengujian Integrasi

Pengujian integrasi merupakan pengujian dari keseluruhan fase pada suatu proses, dimana menggabungkan unit perangkat lunak untuk diuji dan dievaluasi. Terdapat dua kelompok strategi integrasi perangkat lunak, yang pertama *non-incremental* yang terdiri dari pendekatan *big bang*, dan yang kedua *incremental* yang terdiri dari pendekatan *Top-Down testing*, *Bottom-up testing* dan *sandwich*. (Presman, 1997).

2.8.3 Pengujian Validasi

Pengujian validasi ini dilakukan untuk menguji kebutuhan fungsional dari aplikasi *e-learning* menggunakan metode *black box*. Pengujian ini dilakukan untuk mengetahui apakah sistem yang dikembangkan sudah memenuhi kebutuhan yang dibutuhkan dan berfokus pada proses jalannya suatu program. (A. S. & Shalahuddin, 2011).

2.8.4 Pengujian *Usability*

Pengujian *usability* ini digunakan untuk mengetahui apakah pengguna merasa mudah untuk menggunakan aplikasi dari sisi antarmukanya, pengujian *usability* ini terdapat lima kriteria yaitu (Rubin & Chrisnell, 2008):

1. *Learnability*

Untuk mengukur tingkat kemudahan suatu produk.

2. *Efficiency*

Untuk mengukur tingkat performa dalam menggunakan produk.

3. *Satisfaction*

Untuk mengukur apakah pengguna merasa nyaman dan puas dalam menggunakan produk yang dibuat.

2.8.4.1 Tujuan Pengujian *Usability*

Ada beberapa tujuan dari pengujian *usability* sebagai berikut: (Rubin & Chisnell, 2008).

1. Untuk mengetahui kekurangan serta memperbaiki kekurangan yang ada pada sebuah produk yang dibangun.
2. Mencari permasalahan yang terdapat pada produk yang dibuat dan memberbaikinya serta dapat sekaligus melakukan pengembangan pada produk yang dibangun.
3. Memberikan kreasi lebih pada produk yang dibangun sehingga dapat mudah untuk dipelajari cara penggunaan produk yang dibangun serta memberikan kepuasan bagi pengguna serta meningkatkan nilai *usability*-nya.

2.8.4.2 Langkah Pengujian *Usability*

Langkah-langkah untuk melakukan *usability testing* adalah sebagai berikut (Nurhadryani, et al., 2011):

1. Menyiapkan sketsa, *prototype* yang ingin dilakukan pengujian.
2. Merancang alat ukur.

Alat ukur disesuaikan dengan komponen yaitu efektivitas, efisiensi dan kebutuhan pengguna. Pengukuran nilai efektivitas dan efisiensi menggunakan lembar observasi, sedangkan untuk penilaian kepuasan pengguna menggunakan *post-task* kuisioner.

3. Melaksanakan pengujian

Pada saat observasi, responden melakukan langkah-langkah sesuai dengan elemen observasi. Jika responden berhasil maka diberi nilai "Ya" namun jika gagal maka diberi nilai "Tidak".

4. Mengalasi Hasil

Tingkat efektivitas dan efisiensi diukur menggunakan tingkat keberhasilan pengguna dalam menyelesaikan elemen observasi dengan perhitungan sebagai berikut:

$$\text{Efektivitas, Efisiensi (\%)} = \frac{\sum_{i=1}^n x_i}{n} \times 100\% \quad (2.1)$$

Dimana x_i adalah nilai keberhasilan responden ke- i , $x_i = \{0,1\}$.

Sedangkan kepuasan pengguna didapat dari presentase perbandingan antara nilai kepuasan dengan perkalian bobot maksimum skala Likert (5) dengan jumlah responden (n).

$$\text{Kepuasan (\%)} = \frac{\sum_{i=1}^n x_i}{n} \times 100\% \quad (2.2)$$

Dimana x_i adalah nilai keberhasilan responden ke- i , $x_i = \{0,1,2,3,4,5\}$

Sedangkan nilai *usability* sendiri didapat dari rata-rata efektivitas, efisiensi dan kepuasan pengguna.

$$\text{Usability (\%)} = \frac{(\text{efektivitas} + \text{efisiensi} + \text{kepuasan})}{3} \times 100\% \quad (2.3)$$

2.8.4.3 Jumlah Partisipan Pengujian Usability

Untuk melakukan sebuah pengujian *usability* dengan hasil terbaik maka dibutuhkan setidaknya lima orang partisipan tidak peduli dengan perbedaan *platform* (Nielsen, 2012). Dengan menggunakan lima orang partisipan, hasil yang didapatkan akan mendekati rasio maksimum pengujian pengguna.

2.8.4.4 USE Questionnaire

Pada pengujian *usability* menggunakan kuesioner *USE (useful, satisfaction, and ease of use)* untuk mengukur *usefulness, ease of use, ease of learning*, dan *satisfaction* dengan menggunakan skala Likert untuk perhitungan hasilnya. Format kuesioner untuk melakukan pengujian *usability* dapat dilihat pada tabel 2.1 dengan skala Likert sebagai acuan untuk menghitung hasil kuesioner dapat dilihat pada gambar 2.2.

Tabel 2.1 Tabel Kuesioner USE

Sumber: (Lund, 2016)

USEFULLNESS		STS	TS	N	S	SS
1						
2						
EASE OF USE						
3						
4						

EASE OF LEARNING						
5						
6						
SATISFACTION						
7						
8						

Keterangan :

STS = Sangat tidak setuju

TS = Tidak Setuju

N = Netral

S = Setuju

SS = Sangat setuju

Tabel 2.2 Tabel Skala Likert

Sumber : (Bertram, n.d.)

No	Interval	Nilai
1	0% - 19.99%	Sangat Buruk
2	20% - 39.99%	Buruk
3	40% - 59.99%	Cukup
4	60% - 79.99%	Baik
5	80% - 100%	Sangat Baik

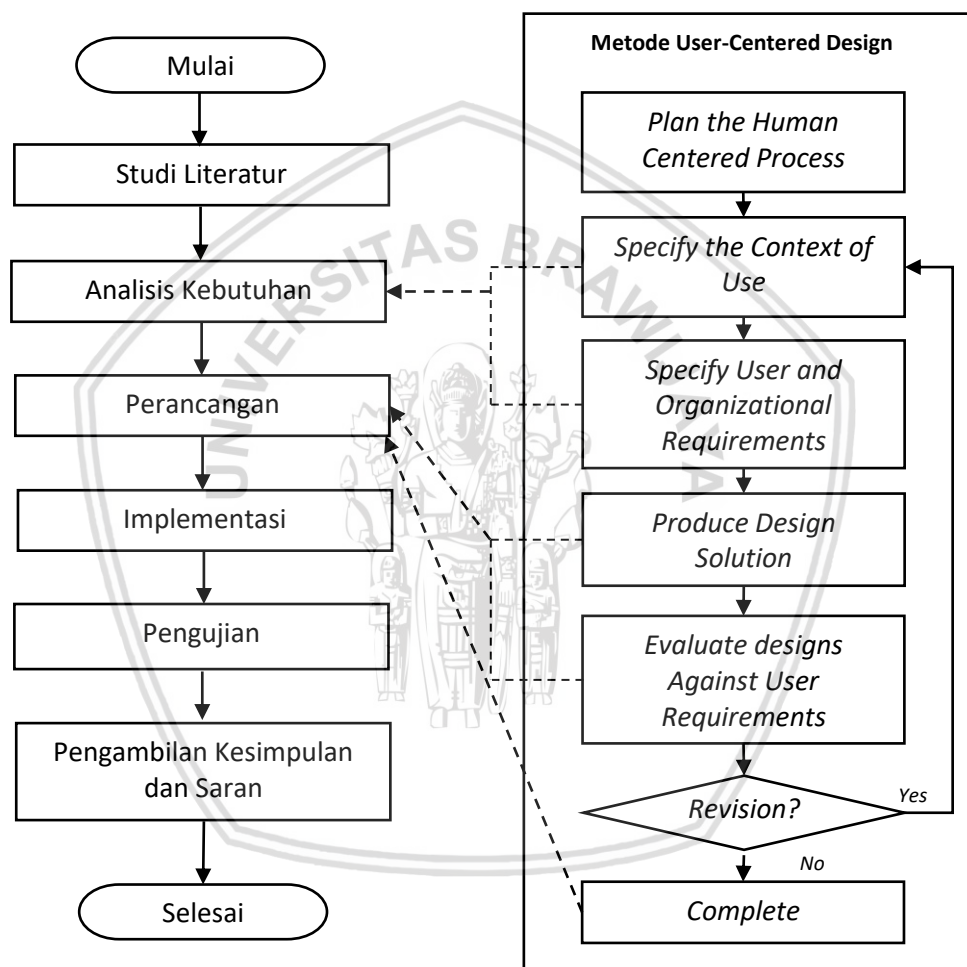
Responden atau sasaran kuesioner tersebut adalah siswa sebagai pengguna utama aplikasi *e-learning* serta beberapa tim pengajar, kemudian setelah mengisi kuesioner akan dilakukan perhitungan dari hasil kuesioner tersebut. (Rianto Rahadi, 2014).

2.8.5 Pengujian *Compatibility*

Pengujian *compatibility* merupakan pengujian yang digunakan untuk mengetahui apakah aplikasi yang dikembangkan mampu berjalan pada berbagai macam *device platform* yang berbeda, versi OS yang berbeda, ukuran layar yang berbeda, serta resolusi layar yang berbeda (Pusuluri, 2006).

BAB 3 METODOLOGI

Pada bab ini penulis membahas tentang metode yang digunakan dalam pembangunan aplikasi *e-learning* sekolah pada *platform* Android untuk menunjang kegiatan pembelajaran untuk siswa dan pengajar. Beberapa tahap yang dilakukan dalam penelitian ini adalah studi literatur, analisis kebutuhan, perancangan perangkat lunak, implementasi, pengujian, dan pengambilan kesimpulan dan saran. Untuk lebih jelasnya metode penelitian ini dapat dilihat pada diagram blok Gambar 3.1 berikut.



Gambar 3.1 Diagram Metodologi Penelitian

3.1 Studi Literatur

Studi literatur digunakan untuk mengetahui penelitian terlebih dahulu yang terkait serta mengetahui dasar-dasar teorinya sebagai acuan ataupun pandangan dalam melakukan penelitian. Informasi dan pustaka yang ada dalam penelitian ini diambil dari berbagai sumber seperti buku, jurnal ilmiah, internet, dan dosen pembimbing. Adapun teori-teori yang peneliti gunakan untuk merealisasikan penelitian ini adalah sebagai berikut:

1. Kajian Pustaka.
2. *E-learning*.
3. Android.
4. Firebase.
5. OneSignal
6. SQLite.
7. *User-Centered Design*.
8. Pengujian Perangkat Lunak.

3.2 Analisis Kebutuhan

Analisis kebutuhan dilakukan peneliti untuk mendapatkan informasi kebutuhan yang diperlukan oleh sistem yang akan dibangun pada aplikasi *e-learning* dengan menggunakan metode *User Centered Design* (UCD). Dengan menggunakan metode UCD pada penelitian ini, ada beberapa alur yang harus dilakukan, yang pertama adalah *specify the context of use*, untuk melakukan identifikasi dan menentukan siapa calon pengguna aplikasi ini. Setelah itu terdapat tahap *specify user and organizational requirement* untuk mengidentifikasi apa saja kebutuhan dari calon pengguna aplikasi, tahap ini dilakukan dengan mengajukan beberapa buah pertanyaan dengan cara wawancara kepada 5 pengguna dari siswa dan 5 pengguna dari pengajar selaku pengguna *e-learning*.

Tahapan selanjutnya yaitu mulai melakukan tahap analisis data, kemudian akan direpresentasikan pada *Use Case Diagram* dan *Use Case Scenario*. Pembuatan *Use Case Diagram* digunakan untuk mengetahui fungsi apa saja yang dibutuhkan oleh sistem dan siapa saja yang menjalankan fungsi-fungsi tersebut. Kemudian untuk alur kerja tiap *Use Case* dijelaskan pada *Use Case Scenario* agar calon pengguna dapat lebih mudah memahami bagaimana cara kerja sistemnya. Dari hasil analisis pada tahap ini menetapkan bahwa terdapat dua aktor yang berperan yaitu aktor siswa dan pengajar, selain itu dari hasil analisis mendapatkan hasil 19 *use case* untuk kedua aktor.

3.3 Perancangan Perangkat Lunak

Tahap perancangan dilakukan setelah selesai melakukan proses analisis kebutuhan. Dengan menggunakan metode UCD maka tahapan tersebut menyesuaikan dengan tahapan pada metode yang digunakan, yaitu pada tahap *product design solution* dan *evaluate design against requirements* yang dilakukan dengan cara membuat *design wireframe* dan dievaluasi oleh calon pengguna secara berkala.

Pada tahap *design solution*, setelah peneliti melewati tahap perancangan arsitektur sistem, perancangan basis data, perancangan *class diagram*, perancangan *diagram sequence*, dan perancangan algoritme, peneliti membuat

perancangan antarmuka. Tahap ini dilakukan untuk memberikan kemudahan bagi pengguna untuk lebih memahami bagaimana produknya nanti dalam bentuk *wireframe*, yang menjelaskan solusi secara umum dengan gambaran perancangan antarmuka yang dibuat dan kerja dari sistem yang akan dibuat secara keseluruhan.

Setelah tahap *design solution* selesai, hasil dari perancangan tersebut diberikan kepada calon pengguna untuk dievaluasi, apakah sudah sesuai atau belum dan disini terjadilah iterasi, jika belum sesuai peneliti harus memperbaiki sistem yang dibuat berdasarkan pada hasil evaluasi yang diberikan oleh calon pengguna tersebut.

Perancangan yang dilakukan ini berdasar pada *object-oriented programming* dengan menggunakan permodelan UML (*Unified Modeling Language*). Perancangan yang dilakukan adalah perancangan arsitektur sistem, perancangan diagram *sequence*, perancangan *class diagram*, perancangan basis data, perancangan kode program dan perancangan antarmuka beserta *screenflow*-nya.

3.4 Implementasi Perangkat Lunak

Setelah semua iterasi pada evaluasi *design prototype* selesai, langkah selanjutnya yang akan diambil adalah tahap implementasi. Tahap ini merupakan tahap dimana perancangan sistem diubah menjadi sesuatu yang benar-benar dapat digunakan oleh pengguna. Implementasi aplikasinya peneliti menggunakan perangkat lunak Android Studio dan Firebase serta SQLite untuk penyimpanan datanya. Semua yang dilakukan berdasar pada tahap perancangan sebelumnya yang telah selesai melalui tahap evaluasi oleh pengguna.

Implementasi diawali dengan mengetahui spesifikasi yang dibutuhkan dalam membangun sistem tersebut, tidak hanya spesifikasi perangkat lunak saja tapi juga spesifikasi perangkat keras. Kemudian juga dijelaskan batasan-batasan masalah yang terjadi saat melakukan implementasi sistem tersebut.

3.5 Pengujian Perangkat Lunak

Tahap ini dilakukan untuk memastikan bahwa aplikasi yang dibangun telah berjalan sesuai dengan yang diinginkan. Pengujian yang dilakukan terdapat beberapa metode yaitu pengujian unit, pengujian integrasi, pengujian validasi, pengujian *usability*, dan pengujian *compatibility*.

1. Pengujian unit dilakukan untuk menguji tiga *method* yang ada pada suatu klas. Pengujian ini dilakukan menggunakan metode *white box*. Jenis pengujian yang dilakukan dengan menggunakan *basis path testing* untuk menguji kode program berdasarkan algoritme pada setiap metode yang ada.
2. Pengujian validasi merupakan pengujian yang digunakan untuk menguji seluruh kebutuhan fungsional pada sistem apakah susah jalan sesuai dengan kebutuhan yang didapatkan sebelumnya. Pengujian ini menggunakan metode *black box*.

3. Pengujian *usability* digunakan untuk mengukur tingkat pengalaman pengguna saat menggunakan sistem ini. Untuk mendapatkan data dari hasil pengujian ini, dilakukan pengisian kuesioner yang diberikan kepada 18 pengguna, diantaranya 14 pengguna siswa dan 4 pengguna pengajar. Untuk soal dan hasil kuisisioner dari pengujian *usability* terlampir pada lampiran.
4. Pengujian *compatibility* digunakan untuk mengetahui apakah aplikasi yang dikembangkan sudah berjalan dengan baik diberbagai *device* siswa dan pengajar yang mayoritas memiliki *device* dengan spesifikasi yang berbeda-beda, namun *device* penguji masih sesuai dengan spesifikasi yang dibutuhkan oleh sistem. Pengujian ini akan dilakukan pada tiga *device* yang berbeda dengan spesifikasi resolusi layar dan versi sistem operasi yang berbeda.

Dengan adanya pengujian maka akan didapatkan segala kekurangan dari sistem yang dibuat dan mengetahui apa saja yang harus diperbaiki untuk kedepannya pada sistem, sehingga isi hasil dari pengujian ini akan dijadikan sebagai isi dari kesimpulan dan saran.

3.6 Pengambilan Kesimpulan dan Saran

Pengambilan simpulan dilakukan setelah selesai melakukan semua tahapan perancangan perangkat lunak, implementasi perangkat lunak serta telah melakukan pengujian perangkat lunak. Kesimpulan didapat dari hasil pengujian terhadap aplikasi yang dibangun. Setelah menyimpulkan kesimpulan, ada tahap penulisan saran dengan guna untuk memperbaiki kesalahan-kesalahan yang terjadi serta menyempurnakan penulisan untuk memberikan pertimbangan atas pengembangan aplikasi selanjutnya.

BAB 4 REKASAYA KEBUTUHAN

Bab ini membahas mengenai analisis sistem pada *e-learning* pada studi kasus SMAN 3 Sidoarjo. Tahapan yang dilakukan peneliti pada bab ini yaitu tahap analisis. Untuk tahap analisis terdapat empat langkah, yaitu gambaran umum sistem, mengidentifikasi aktor yang terlibat pada sistem, menganalisis kebutuhan pengguna baik kebutuhan fungsional dan kebutuhan non-fungsional.

4.1 Analisis Kebutuhan

Pada tahap analisis kebutuhan menerapkan metode UCD pada tahap *specify the context of use* dan *specify user and organizational requirements* yang berarti pada tahap ini dilakukan proses tentang gambaran umum sistem, identifikasi aktor, serta penjabaran kebutuhan fungsional maupun kebutuhan non-fungsional. Analisis kebutuhan ini memiliki tujuan untuk menggambarkan kebutuhan-kebutuhan apa saja yang harus ada pada sistem agar dapat memenuhi kebutuhan pengguna.

4.1.1 Gambaran Umum Aplikasi

Pada bahasan gambaran umum sistem *e-learning* pada studi kasus SMAN 3 Sidoarjo ini dijabarkan pada deskripsi aplikasi.

4.1.1.1 Deskripsi Aplikasi

Aplikasi yang dibangun dengan nama “E-learning SMANTIGDA” menunjukkan bahwa aplikasi yang dibuat ini merupakan sebuah elearning dari salah satu sekolah menengah atas di Sidoarjo yaitu SMA Negeri 3 Sidoarjo. Tujuan utama dari pembangunan sistem ini adalah sebagai media penunjang kegiatan pembelajaran serta sarana atau media informasi yang lebih cepat tersalurkan untuk siswa dan pengajar SMAN 3 Sidoarjo. Penggunaanya baik dari siswa maupun pengajar diharapkan dapat berperan lebih aktif lagi dalam bidang pendidikan dengan cara mengakses materi mata pelajaran, informasi tugas dan segala informasi tentang sekolah dengan mudah dan cepat.

Ada beberapa fitur utama yang dimiliki, yaitu jadwal pelajaran untuk memudahkan siswa dan pengajar melihat jadwal pelajaran yang sedang diterapkan oleh sekolah. Jadwal pelajaran dibagi tiap kelas untuk mempermudah pencarian jadwal sesuai dokumen resmi yang dibuat oleh sekolah. Selain itu juga ada materi pelajaran yang telah dikelompokkan menjadi beberapa sub menu kelas dan keminatan yang nantinya akan dapat membuka *ebook* resmi yang disediakan oleh Kementerian Pendidikan dan Kebudayaan Republik Indonesia. Kemudian terdapat fitur tugas untuk melihat daftar tugas yang telah diberikan oleh pengajar, serta dapat melakukan pengumpulan tugas langsung dari *smartphone* dengan format yang telah ditentukan. *Timeline* kalender akademik untuk melihat *timeline* akademik di dunia pendidikan selama satu tahun ajaran baru. Fitur aspirasi untuk menyampaikan aspirasi yang ditujukan pada pihak sekolah termasuk siswa, pengajar, ekstrakurikuler, laboratorium dan lainnya.

E-learning ini memiliki suatu lingkaran agar sistem dapat berjalan. Sistem ini memerlukan sebuah perangkat berupa *smartphone* dengan sistem operasi Android sebagai media berjalannya sistem. Perangkat ini dipilih karena banyaknya siswa dan pengajar SMAN 3 Sidoarjo merupakan pengguna *smartphone* ber-sistem operasi Android. Selain itu tingginya tingkat mobilitasnya yang tinggi menyebabkan sistem dapat diakses kapan saja dan dimana saja.

4.1.1.2 Cara Penggunaan Sistem

Aplikasi *e-learning* ini terdapat beberapa fitur utama antara Jadwal Pelajaran, Materi Pelajaran, Tugas, Kalender Akademik, Aspirasi, Akun dan Pendaftaran. Pengguna menggunakan fitur-fitur tersebut sesuai kebutuhan yang akan pengguna lakukan. Ada beberapa fitur yang mengharuskan pengguna terhubung dengan jaringan internet untuk dapat menggunakan beberapa fitur pada aplikasi *e-learning* tersebut.

4.1.2 Identifikasi Aktor

Tahap identifikasi aktor diperlukan untuk dapat menentukan aktor-aktor yang akan berinteraksi dengan sistem. Sistem ini memungkinkan seluruh pengguna baik siswa maupun tim pengajar dapat bertukar informasi untuk menunjang kegiatan ajar-mengajar di SMAN 3 Sidoarjo. Tahap ini dilakukan dengan menjabarkan aktor-aktor yang menjadi pengguna dalam sistem beserta deskripsinya. Daftar aktor beserta deskripsinya dapat dilihat pada Tabel 4.1.

Tabel 4.1 Identifikasi Aktor

Aktor	Deskripsi
Pengajar	Pengajar adalah pengguna sistem yang dapat menggunakan fitur-fitur yang telah disediakan oleh aplikasi dari sisi siswa, antara lain melihat jadwal pelajaran serta menambahkan informasi pergantian jadwal, materi pelajaran, lihat dan tugas, lihat dan tambah kegiatan akademik, kirim aspirasi serta pengajar dapat melakukan pendaftaran akun siswa.
Siswa	Siswa adalah pengguna sistem yang dapat menggunakan fitur-fitur yang disediakan oleh aplikasi dari sisi siswa, antara lain melihat jadwal pelajaran dan lihat informasi perubahan jadwal, lihat materi pelajaran, lihat daftar tugas dan unggah tugas, lihat timeline akademik, dan fitur untuk menyuarakan aspirasi.

4.2 Analisis Data

Aplikasi ini membutuhkan beberapa data dari pengguna baik siswa ataupun pengajar. Berikut adalah data yang diperlukan oleh sistem:

1. Data pengguna berupa *email* dan *password* yang digunakan untuk masuk ke sistem.
2. Data siswa berupa nama lengkap dan tempat tanggal lahir untuk data tambahan siswa, *password* pada setiap akun siswa, serta berkas tugas.

3. Data pengajar berupa informasi perubahan jadwal, informasi tugas untuk siswa, informasi kegiatan.

4.3 Analisis Kebutuhan Fungsional

Analisis kebutuhan fungsional merupakan proses analisis untuk mengetahui fungsi apa saja yang harus dipenuhi serta berisi proses-proses apa saja yang dilakukan oleh sistem. Mengetahui kebutuhan fungsional ini dilakukan dengan menggunakan metode teknik *Probably Sampling* dengan jenis sampling *Simple Random Sampling* yang mengambil calon pengguna aplikasi secara acak wawancara langsung terhadap 10 calon pengguna aplikasi, 5 calon pengguna dari siswa dan 5 calon pengguna dari pengajar yang telah ditetapkan sebelumnya (Margono, 2004). Berikut daftar kebutuhan berdasarkan kebutuhan fungsional yang harus dimiliki sistem dapat dilihat pada Tabel 4.2 dan Tabel 4.3.

Tabel 4.2 Rekap Kebutuhan Pengguna (Siswa)

Nama Kebutuhan	Jumlah Orang	Prosentase
1. Pengguna membutuhkan sarana penunjang kegiatan pembelajaran?	Setuju: 5 Orang Tidak setuju: 0 Orang	100%
2. Pengguna membutuhkan informasi jadwal pelajaran?	Setuju: 5 Orang Tidak setuju: 0 Orang	100%
3. Pengguna membutuhkan fitur materi pelajaran?	Setuju: 5 Orang Tidak setuju: 0 Orang	100%
4. Pengguna membutuhkan fitur Tugas untuk mengetahui daftar tugas?	Setuju: 5 Orang Tidak setuju: 0 Orang	100%
5. Pengguna membutuhkan informasi timeline kegiatan akademik?	Setuju: 5 Orang Tidak setuju: 0 Orang	100%

Tabel 4.3 Rekap Kebutuhan Pengguna (Pengajar)

Nama Kebutuhan	Jumlah Orang	Prosentase
1. Pengguna membutuhkan sarana penunjang kegiatan pembelajaran?	Setuju: 5 Orang Tidak setuju: 0 Orang	100%
2. Pengguna membutuhkan informasi jadwal pelajaran?	Setuju: 5 Orang Tidak setuju: 0 Orang	100%
3. Pengguna membutuhkan fitur materi pelajaran?	Setuju: 5 Orang	100%

	Tidak setuju: 0 Orang	
4. Pengguna membutuhkan fitur tugas untuk dikirimkan kepada siswa?	Setuju: 5 Orang Tidak setuju: 0 Orang	100%
5. Pengguna membutuhkan informasi timeline kegiatan akademik?	Setuju: 5 Orang Tidak setuju: 0 Orang	100%

Dari beberapa responden wawancara peneliti dapat merumuskan sepuluh fungsi. Semua fungsi tersebut harus dimiliki oleh sistem karena berhubungan dengan fitur-fitur yang dimiliki oleh sistem. Untuk fitur UTS, UAS dan ulangan harian sampai saat ini siswa tidak merasa terganggu dan masih merasa nyaman saat melakukan kegiatan tersebut pada *e-learning* sekolah saat ini, maka fitur tersebut tidak akan dibuat pada aplikasi yang akan dibangun, namun terdapat fitur sejenis yaitu fitur latihan soal untuk melakukan latihan-latihan soal persiapan menghadapi ujian dan ulangan harian. Daftar kebutuhan fungsional tersebut akan dijabarkan dalam diagram *use case*. Kebutuhan fungsional pertama untuk sisi siswa yang bersifat sementara dapat dilihat pada Tabel 4.4.

Tabel 4.4 Kebutuhan Fungsional Siswa

No	Kode Fungsi	Nama Fungsi	Deskripsi
1	ESS-F-01	Daftar	Siswa melakukan pendaftaran agar dapat menggunakan sistem
2	ESS-F-02	Masuk	Siswa dapat melakukan <i>login</i> agar dapat masuk ke dalam sistem
3	ESS-F-03	Lihat Kalender Akademik	Siswa memilih menu Kalender akademik maka sistem akan menampilkan kegiatan akademik sekolah tiap tahunnya.
4	ESS-F-04	Lihat Materi Pelajaran	Siswa memilih menu Materi Pelajaran maka sistem akan menampilkan materi pelajaran yang dikategorikan berdasarkan kelas dan keminatan.
5	ESS-F-05	Lihat Jadwal Pelajaran	Siswa memilih menu Jadwal Pelajaran maka sistem akan menampilkan pilihan kelas yang akan ditampilkan jadwal pelajarannya.
6	ESS-F-06	Lihat Tugas	Siswa memilih menu tugas maka sistem akan menampilkan daftar tugas yang diberikan oleh tim

			pengajar untuk siswa.
7	ESS-F-07	Kirim Kritik Saran	Siswa dapat memberikan kritik dan saran pada pihak sekolah melalui menu kotak saran.
8	ESS-F-08	Lihat Akun detil	Siswa dapat melihat profil akun pengguna
9	ESS-F-09	Ubah akun	Siswa dapat merubah profil akun sesuai identitasnya
10	ESS-F-10	Keluar	Siswa melakukan <i>logout</i> untuk dapat keluar dari sistem

Setelah dilakukannya evaluasi pertama oleh calon pengguna, berikut hasil yang menyatakan perlu dilakukan adanya perubahan pada daftar kebutuhan fungsional seperti pada Tabel 4.5.

Tabel 4.5 Kebutuhan Fungsional Siswa (Iterasi 1)

No	Kode Fungsi	Nama Fungsi	Deskripsi
1	ESS-F-07	Kirim Aspirasi	Siswa dapat menyuarakan aspirasinya secara anonim untuk ditujukan kepada salah satu atau beberapa civitas sekolah.
2	ESS-F-11	Lihat Aspirasi	Siswa melihat daftar aspirasi yang telah dimasukkan sebelumnya oleh akun pengajar maupun akun siswa.
3	ESS-F-12	Unggah Tugas	Siswa dapat melakukan aktivitas unggah tugas untuk mengumpulkan tugas yang telah dikerjakan.

Daftar kebutuhan fungsional terdapat beberapa perubahan, seperti pada nomor kebutuhan ES-F-01 untuk dihapus karena untuk mengurangi data *noise* atau bukan civitas sekolah yang menggunakan sistem ini. Selain itu juga dapat mempercepat proses penggunaan sistem oleh para siswa karena data siswa telah didaftarkan terlebih dahulu pada sistem saat sistem dibuat.

Juga terdapat perubahan nama fungsi dari kode ES-F-07 disesuaikan dengan nama sarana yang disediakan di sekolah untuk menyampaikan aspirasinya kepada pihak sekolah. Serta penambahan fungsi dengan kode ES-F10 untuk melihat daftar aspirasi yang telah dikirimkan oleh pengguna aplikasi.

Kemudian juga ada beberapa penambahan daftar kebutuhan, yaitu ES-F-11 dengan nama fungsi unggah tugas. Adanya penambahan fungsi ini dikarenakan banyaknya siswa mengeluhkan tidak adanya lokasi terpusat untuk pengumpulan

tugas yang sering kali menyusahkan siswa untuk mencari *email* tujuan atau media penyimpanan yang sering berganti lokasi.

Setelah dilakukannya evaluasi kedua oleh calon pengguna, berikut hasil yang menyatakan perlu dilakukan adanya perubahan pada daftar kebutuhan fungsional, seperti pada Tabel 4.6.

Tabel 4.6 Kebutuhan Fungsional Siswa (Iterasi 2)

No	Kode Fungsi	Nama Fungsi	Deskripsi
1	ESS-F-03	Lihat <i>Timeline</i> Akademik	Siswa memilih menu <i>Timeline</i> akademik maka sistem akan menampilkan kegiatan akademik sekolah tiap tahunnya.
2	ESS-F-13	Lihat Info Perubahan Jadwal	Siswa melihat daftar perubahan jadwal pada menu jadwal pelajaran.
3	ESS-F-14	Ubah <i>Password</i>	Siswa dapat mengubah <i>password</i> pengguna

Setelah mengalami beberapa perubahan pada kode fungsi ESS-F-07 dan penambahan kode fungsi ESS-F-11 dan ESS-F-12. Terdapat perubahan dan penambahan pada kebutuhan fungsionalnya. Pada kode fungsi ESS-F-03 terdapat perubahan nama fungsi menjadi Lihat *Timeline* Akademik, karena isi dari menu tersebut bukanlah kalender melainkan berbentuk daftar seperti *timeline*. Selain itu juga terdapat penambahan kebutuhan fungsional lagi yaitu dengan kode fungsi ESS-F-13 dengan nama fungsi Lihat Info Perubahan Jadwal yang digunakan oleh pengguna untuk melihat informasi jadwal pelajaran yang terkadang sering terjadi perubahan sewaktu-waktu. Terakhir pada kode fungsi ESS-F-14 digunakan untuk melakukan ubah password pengguna.

Untuk kebutuhan fungsional dari sisi pengajar, dari beberapa responden wawancara peneliti dapat merumuskan tiga belas fungsi. Kebutuhan fungsional pertama dari sisi pengajar yang bersifat sementara dapat dilihat pada Tabel 4.7.

Tabel 4.7 Kebutuhan Fungsional Pengajar

No	Kode Fungsi	Nama Fungsi	Deskripsi
1	ESP-F-01	Daftar	Pengajar dapat melakukan daftar agar dapat masuk kedalam sistem
2	ESP-F-02	Masuk	Pengajar dapat melakukan login agar dapat masuk kedalam sistem
3	ESP-F-03	Lihat Kalender Akademik	Pengajar memilih menu Kalender akademik maka sistem akan menampilkan kegiatan akademik sekolah tiap tahunnya.

4	ESP-F-04	Lihat Materi Pelajaran	Pengajar memilih menu Materi Pelajaran maka sistem akan menampilkan materi pelajaran yang dikategorikan berdasarkan kelas dan keminatan.
5	ESP-F-05	Lihat Jadwal Pelajaran	Pengajar memilih menu Jadwal Pelajaran maka sistem akan menampilkan pilihan kelas yang akan ditampilkan jadwal pelajarannya.
6	ESP-F-06	Lihat Tugas	Pengajar memilih menu tugas maka sistem akan menampilkan daftar tugas yang diberikan oleh tim pengajar untuk siswa.
7	ESP-F-07	Tambah Tugas	Pengajar dapat menambah daftar tugas untuk dikirimkan kepada siswa
8	ESP-F-08	Lihat Berkas Tugas	Pengajar memilih menu lihat berkas untuk melihat direktori pengumpulan tugas dari siswa
9	ESP-F-09	Kirim Kritik Saran	Pengajar dapat memberikan kritik dan sarannya secara anonim untuk ditujukan kepada salah satu atau beberapa civitas sekolah
10	ESP-F-10	Lihat Daftar Kritik Saran	Pengajar melihat daftar kritik dan saran yang telah dimasukkan sebelumnya oleh akun pengajar maupun akun siswa
11	ESP-F-11	Tambah Akun	Pengajar menambah akun pengguna sistem
12	ESP-F-12	Ubah Akun	Pengajar melakukan pembaharuan data akun
13	ESP-F-13	Keluar	Pengajar melakukan aktivitas <i>logout</i> untuk dapat keluar dari sistem

Kebutuhan fungsional terdapat beberapa perubahan, pada nomor kebutuhan ESP-F-01 untuk dihapus karena untuk mengurangi data *noise* atau bukan civitas sekolah yang menggunakan sistem ini. Selain itu juga dapat mempercepat proses penggunaan sistem oleh para pengajar karena data pengajar telah didaftarkan terlebih dahulu pada sistem saat sistem dibuat.

Juga terdapat perubahan nama fungsi dari kode ESP-F-09 disesuaikan dengan nama sarana yang disediakan di sekolah untuk menyampaikan aspirasinya kepada pihak sekolah. Serta penambahan fungsi dengan kode ESP-F10 untuk melihat daftar aspirasi yang telah dikirimkan oleh pengguna aplikasi.

Terdapat perubahan nama fungsi untuk kode fungsi ESP-F-11, fungsi pendaftaran ini hanya dapat dilakukan oleh pegajar. Penambahan fungsi ini untuk mempermudah dalam penggunaan sistem jika terdapat pengguna baru baik dari siswa saja. Perubahan kebutuhan fungsional pelajar yang terjadi dijelaskan pada tabel 4.8.

Tabel 4.8 Kebutuhan Fungsional Pengajar (Iterasi 1)

No	Kode Fungsi	Nama Fungsi	Deskripsi
1	ESP-F-09	Kirim Aspirasi	Siswa dapat menyuarakan aspirasinya secara anonim untuk ditujukan kepada salah satu atau beberapa civitas sekolah.
2	ESP-F-10	Lihat Aspirasi	Pengajar melihat daftar aspirasi yang telah dimasukkan sebelumnya oleh akun pengajar maupun akun siswa.
3	ESP-F-14	Mendaftar akun siswa baru	Pengajar melakukan pendaftaran akun siswa agar dapat menggunakan sistem.

Setelah adanya penambahan dan perubahan kebutuhan fungsional aktor pengajar pada beberapa kode fungsi, calon pengguna masih meminta dilakukannya perubahan pada kebutuhan fungsional pada sisi pengajar. Yaitu kode fungsi ESP-F-03 terjadi perubahan nama fungsi untuk menyesuaikan fungsinya, kemudian juga terdapat penambahan kebutuhan fungsional dengan kode fungsi ESP-F-15 untuk melihat info perubahan jadwal pelajaran, kemudian kode fungsi ESP-F-16 untuk menambah kegiatan pada menu timeline akademik, kemudian kode fungsi ESP-F-17 untuk melakukan penambahan info perubahan jadwal pelajaran yang dapat berubah sewaktu-waktu. Dan yang terakhir pada kode fungsi ESP-F-18 untuk melakukan unduh berkas tuags yang terdapat pada laman lihat berkas.

Tabel 4.9 Kebutuhan Fungsional Pengajar (Iterasi 2)

No	Kode Fungsi	Nama Fungsi	Deskripsi
1	ESP-F-03	Lihat <i>Timeline</i> Akademik	Pengajar memilih menu <i>Timeline</i> akademik maka sistem akan menampilkan kegiatan akademik sekolah tiap tahunnya.

2	ESP-F-15	Lihat Info Perubahan Jadwal	Pengajar melihat daftar perubahan jadwal pada menu jadwal pelajaran.
3	ESP-F-16	Tambah Kegiatan	Pada menu <i>Timeline</i> Akademik pengajar menambahkan kegiatan baru agar dapat dilihat oleh siswa.
4	ESP-F-17	Tambah info perubahan jadwal	Pengajar dapat menambahkan info baru jika terdapat perubahan jadwal pelajaran sewaktu-waktu.
5	ESP-F-18	Unduh Tugas	Pengajar dapat melakukan aktivitas unduh pada daftar tugas yang ada

4.4 Analisis Kebutuhan Non-Fungsional

Selain terdapat kebutuhan fungsional, pada aplikasi ini juga terdapat kebutuhan non-fungsional. Kebutuhan non-fungsional ini bertujuan untuk mendapatkan spesifikasi tentang aspek kualitas yang dibutuhkan pada aplikasi. Banyaknya jenis *device* dengan spesifikasi yang berbeda-beda yang digunakan oleh siswa dan pengajar, kebutuhan non-fungsional *compatibility* sangat dibutuhkan pada penelitian ini. Kemudian kebutuhan non-fungsional *usability* juga sangat dibutuhkan untuk mengetahui apakah aplikasi yang dikembangkan telah memenuhi parameter *usability* agar aplikasi yang dikembangkan oleh pengguna. Kemudian Hasil analisis kebutuhan non-fungsional aplikasi ini dijelaskan pada Tabel 4.10.

Tabel 4.10 Daftar Kebutuhan non-Fungsional

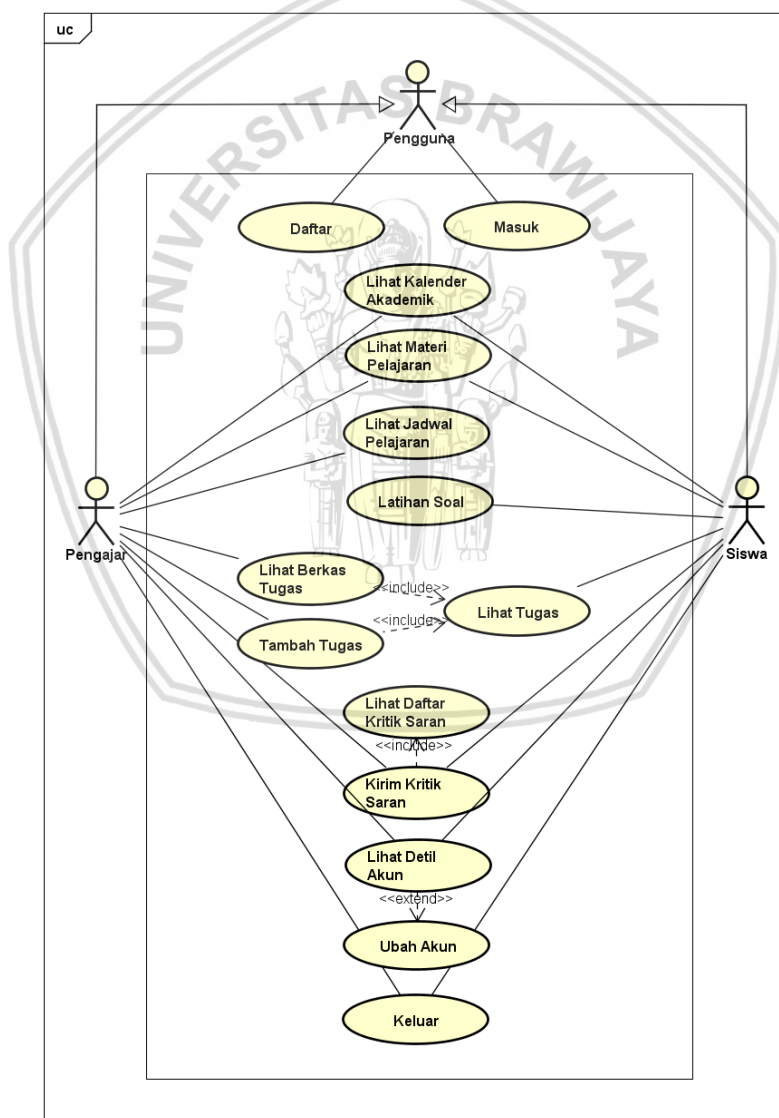
Kebutuhan non-Fungsional	Deskripsi
<i>Usability</i>	Aplikasi dapat memenuhi kebutuhan <i>usability</i> dengan parameter <i>learnability</i> , <i>efficiency</i> , <i>memorability</i> , <i>errors</i> dan <i>satisfaction</i> .
<i>Compatibility</i>	Aplikasi dapat memenuhi kebutuhan <i>compatibility</i> dengan melakukan pengujian aplikasi yang dicoba pada beberapa perangkat sesuai spesifikasi aplikasi yang sedang dikembangkan.

4.5 Diagram Use Case

Diagram ini digunakan untuk memodelkan perilaku sistem yang berkaitan dengan kebutuhan fungsional. Diagram *use case* akan menunjukkan aktor-aktor mana saja yang terlibat dalam sistem serta menunjukkan hubungan antara aktor dengan kebutuhan fungsional yang ada.

Diagram *use case* sistem *e-learning* dapat dilihat pada Gambar 4.1 menunjukkan fungsi-fungsi yang harus ada dalam sistem. Dalam diagram tersebut terdapat tiga aktor yaitu pengguna, siswa dan pengajar. Pada diagram *use case* dapat terlihat bahwa aktor siswa dan pengajar merupakan aktor turunan dari pengguna karena aktor tersebut berkedudukan sama sebagai *user* sebelum melakukan aktivitas *login*. Sistem akan mengenali *level* aktor ketika pengguna sudah melakukan aktivitas *login*.

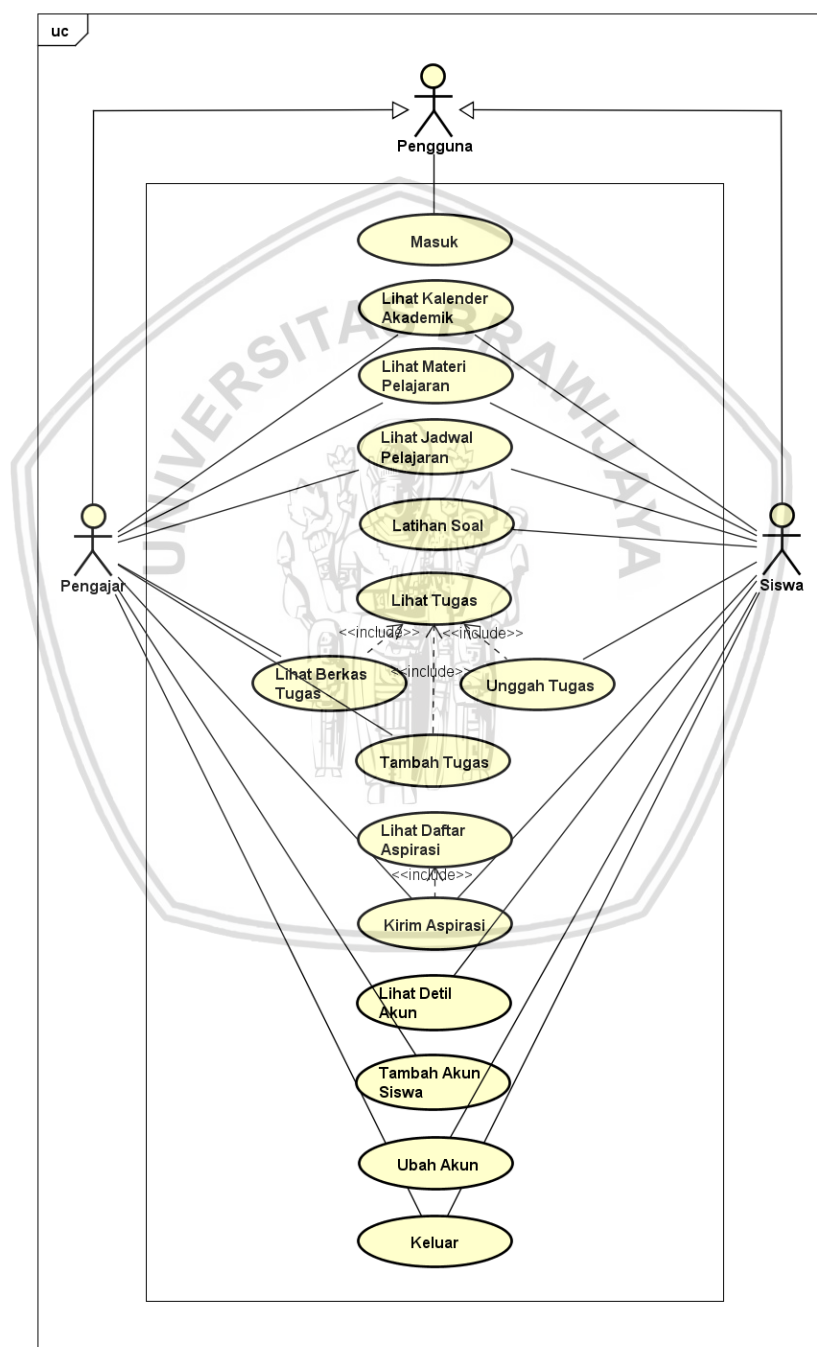
Untuk aktor siswa berhubungan dengan sembilan *use case* yaitu lihat kalender akademik, lihat materi pelajaran, lihat jadwal pelajaran, latihan soal, lihat tugas, kirim kritik saran, lihat detail akun, dan keluar. Untuk aktor pengajar berhubungan dengan delapan *use case* yaitu lihat kalender akademik, lihat materi pelajaran, lihat jadwal pelajaran, lihat berkas tugas, tambah tugas, kirim kritik saran, lihat detail akun, dan keluar.



Gambar 4.1 Diagram Use Case

4.6 Diagram Use Case (Iterasi 1)

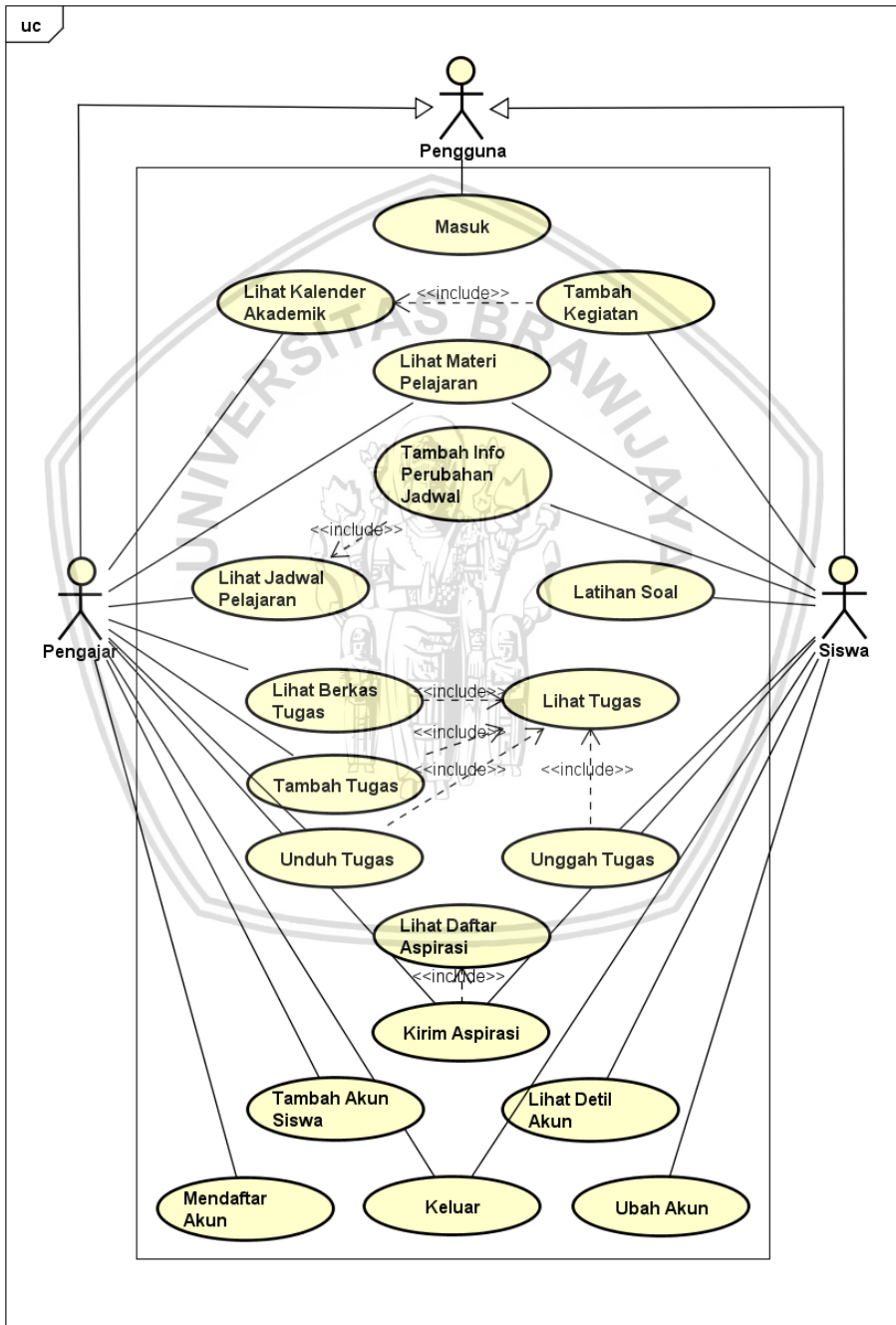
Pada *Use case* iterasi 1 ini terdapat beberapa perubahan dari diagram *use case* sebelumnya, yaitu dihapusnya *use case* diagram daftar, pergantian nama fungsi dari kirim kritik saran dan lihat kritik saran menjadi kirim aspirasi dan lihat aspirasi. Kemudian juga terdapat dua *use case* baru yaitu *use case* unggah tugas untuk aktor siswa dan *use case* mendaftarkan akun siswa baru. Diagram *use case* iterasi 1 dapat dilihat pada Gambar 4.2.



Gambar 4.2 Diagram Use Case (Iterasi 1)

4.7 Diagram Use Case (Iterasi 2)

Use case iterasi 2 juga terdapat beberapa perubahan dan penambahan use case. Terdapat use case baru yaitu lihat info perubahan jadwal untuk kedua aktor, dan tambah kegiatan untuk aktor pengajar. Kemudian ada perubahan nama use case juga yaitu berubahnya use case lihat kalender akademik diubah menjadi lihat timeline akademik. Diagram use case iterasi 2 dapat dilihat pada Gambar 4.3.



Gambar 4.3 Diagram Use Case (Iterasi 2)

4.8 Skenario Use Case

Setiap *use case* diagram akan dijabarkan dan dideskripsikan lebih lanjut tentang perilakunya dalam skenario *use case*. Skenario *use case* ini berisi nama *use case*, aktor yang terlibat pada *use case* tersebut, tujuan dari *use case*, kondisi awal yang harus dipenuhi, alur utama *use case*, alur alternatif, dan kondisi akhir yang diharapkan. Skenario *use case* akan menjadi dasar pada perancangan diagram aktivitas dan diagram sekuen.

4.8.1 Skenario Use Case Masuk (Siswa)

Kebutuhan fungsional masuk diperlukan agar pengguna dapat masuk pada sistem. Pengguna dapat masuk menggunakan email dan kata sandi yang telah didaftarkan sebelumnya. Aktivitas masuk ini diperukan agar pengguna dapat mengakses fitur-fiitur yang dimiliki oleh sistem. Berikut scenario *use case* masuk pada Tabel 4.11.

Tabel 4.11 Skenario Use Case Masuk

<i>Objective</i>	Pengguna melakukan aktivitas masuk untuk masuk ke dalam sistem
<i>Actor</i>	Siswa
<i>Pre-condition</i>	Pengguna belum melakukan aktivitas masuk
<i>Main flow</i>	<ol style="list-style-type: none"> 1. Pengguna memasukkan data email dan kata sandi, pada halaman masuk. 2. Sistem akan melakukan validasi masukan email dan kata sandi pada <i>database</i>.
<i>Alternative flows</i>	Jika masukan salah atau kosong, akan muncul peringatan "Login gagal"
<i>Post-condition</i>	Pengguna masuk ke dalam sistem dan dapat melakukan aktivitas di dalam aplikasi tersebut.

4.8.2 Skenario Use Case Masuk (Pengajar)

Kebutuhan fungsional masuk diperlukan agar pengguna dapat masuk pada sistem. Pengguna dapat masuk menggunakan NIP untuk melakukan validasi, kemudian memasukkan email dan kata sandi yang telah didaftarkan sebelumnya. Aktivitas masuk ini diperukan agar pengguna dapat mengakses fitur-fiitur yang dimiliki oleh sistem. Berikut scenario *use case* masuk pada Tabel 4.12

Tabel 4.12 Skenario Use Case Masuk

<i>Objective</i>	Pengguna melakukan aktivitas masuk untuk masuk ke dalam sistem
<i>Actor</i>	Pengajar

<i>Pre-condition</i>	Pengguna belum melakukan aktivitas masuk
<i>Main flow</i>	<ol style="list-style-type: none"> 1. Pengguna memasukkan NIP 2. Pengguna memasukan data email dan kata sandi, pada halaman masuk. 3. Sistem akan melakukan validasi masukan email dan kata sandi pada <i>database</i>.
<i>Alternative flows</i>	Jika masukan salah atau kosong, akan muncul peringatan "Login gagal"
<i>Post-condition</i>	Pengguna masuk ke dalam sistem dan dapat melakukan aktivitas di dalam aplikasi tersebut.

4.8.3 Skenario Use Case Lihat Timeline Kegiatan Akademik

Kebutuhan selanjutnya adalah kebutuhan *Timeline* Akademik, fungsionalitas ini berfungsi untuk menampilkan *timeline* kegiatan akademik sekolah yang berisikan tentang informasi-informasi kegiatan sekolah dalam jangka waktu tertentu. Berikut skenario *use case* Kalender akademik dapat dilihat pada Tabel 4.13.

Tabel 4.13 Skenario Use Case Lihat Timeline Akademik

<i>Objective</i>	Pengguna memilih menu Timeline Akademik untuk menampilkan kegiatan sekolah
<i>Actor</i>	Siswa, Pengajar.
<i>Pre-condition</i>	Pengguna memilih menu Kalender Akademik
<i>Main flow</i>	<ol style="list-style-type: none"> 1. Pengguna memilih menu kalender akademik 2. Aplikasi akan menampilkan timeline kalender akademik pada jangka waktu tertentu
<i>Alternative flows</i>	Jika tidak terdapat koneksi internet maka sistem tidak dapat menampilkan daftar kegiatan akademik
<i>Post-condition</i>	Pengguna melihat daftar kegiatan sekolah dalam jangka waktu tertentu.

4.8.4 Skenario Use Case Tambah Kegiatan Akademik

Kebutuhan selanjutnya adalah kebutuhan *Timeline* Akademik, fungsionalitas ini berfungsi untuk menampilkan *timeline* kegiatan akademik sekolah yang berisikan tentang informasi-informasi kegiatan sekolah dalam jangka waktu tertentu. Berikut skenario *use case* Kalender akademik dapat dilihat pada Tabel 4.14.

Tabel 4.14 Skenario Use Case Tambah Kegiatan Akademik

<i>Objective</i>	Pengguna memilih menu <i>Timeline</i> Akademik dan menekan tombol tambah kegiatan untuk menambah daftar kegiatan akademik
<i>Actor</i>	Pengajar.
<i>Pre-condition</i>	Pengajar menekan tombol tambah kegiatan pada menu timeline akadeik
<i>Main flow</i>	<ol style="list-style-type: none"> 1. Pengguna memilih menu <i>Timeline</i> akademik 2. Aplikasi akan menampilkan halaman <i>form</i> untuk menambah kegiatan akademik 3. Tekan tombol tambah untuk menambah kegiatan akadmeik
<i>Alternative flows</i>	Jika tidak terdapat koneksi internet maka sistem tidak dapat menyimpan kegiatan akademik baru
<i>Post-condition</i>	Pengguna melihat kegiatan sekolah baru dalam daftar

4.8.5 Skenario Use Case Lihat Jadwal Pelajaran

Skenario *Use Case* Jadwal pelajaran ini berfungsi agar pengguna dapat melihat jadwal pelajaran yang dibagi berdasarkan kelas yang ada pada sekolah. Berikut *use case* jadwal pelajaran dapat dilihat pada Tabel 4.15.

Tabel 4.15 Skenario Use Case Lihat Jadwal Pelajaran

<i>Objective</i>	Pengguna memilih menu Jadwal Pelajaran untuk menampilkan jadwal pelajaran siswa
<i>Actor</i>	Siswa, Pengajar
<i>Pre-condition</i>	Pengguna memilih menu Jadwal Pelajaran
<i>Main flow</i>	<ol style="list-style-type: none"> 1. Pengguna memilih menu Jadwal Pelajaran 2. Aplikasi akan menampilkan pilihan kelas pada <i>actionbar</i> 3. <i>Main Layout</i> akan menampilkan dokumen jadwal pelajaran
<i>Alternative flows</i>	Jika dokumen jadwal pelajaran tidak tersedia, maka sistem akan menampilkan halaman kosong pada <i>main layout</i>
<i>Post-condition</i>	Pengguna melihat jadwal pelajaran

4.8.6 Skenario *Use Case* Lihat Info Perubahan Jadwal Pelajaran

Skenario *Use Case* Lihat Info perubahan Jadwal pelajaran ini berfungsi agar pengguna dapat mengetahui informasi jika terdapat perubahan jadwal yang terjadi. Berikut *use case* jadwal pelajaran dapat dilihat pada Tabel 4.16.

Tabel 4.16 Skenario *Use Case* Lihat Info Perubahan Jadwal Pelajaran

<i>Objective</i>	Pengguna memilih menu Jadwal Pelajaran untuk menampilkan jadwal pelajaran siswa
<i>Actor</i>	Siswa, Pengajar
<i>Pre-condition</i>	Pengguna memilih menu Jadwal Pelajaran
<i>Main flow</i>	<ol style="list-style-type: none"> 1. Pengguna memilih menu Jadwal Pelajaran 2. Aplikasi akan menampilkan info perubahan jadwal pada bagian atas
<i>Alternative flows</i>	Jika tidak terdapat koneksi internet, info perubahan jadwal tidak akan muncul
<i>Post-condition</i>	Pengguna melihat info perubahan jadwal

4.8.7 Skenario *Use Case* Tambah Info Perubahan Jadwal Pelajaran

Skenario *Use Case* Tambah Info Perubahan Jadwal pelajaran ini berfungsi agar pengajar dapat menambah daftar info perubahan jadwal baru. Berikut *use case* info perubahan jadwal pelajaran dapat dilihat pada Tabel 4.17.

Tabel 4.17 Skenario *Use Case* Tambah Info Perubahan Jadwal Pelajaran

<i>Objective</i>	Pengguna memilih menu Jadwal Pelajaran untuk menampilkan jadwal pelajaran siswa
<i>Actor</i>	Pengajar
<i>Pre-condition</i>	Pengguna memilih menu Jadwal Pelajaran
<i>Main flow</i>	<ol style="list-style-type: none"> 1. Pengguna memilih menu Jadwal Pelajaran 2. Pengguna menekan tombol tambah info perubahan jadwal 3. Aplikasi akan menampilkan halaman form perubahan jadwal 4. Pengguna menekan tombol tambah untuk menambah daftar info perubahan jadwal baru
<i>Alternative flows</i>	Jika tidak terdapat koneksi internet, info perubahan jadwal tidak akan tersimpan pada <i>database</i> dan tidak akan muncul dalam daftar info perubahan jadwal
<i>Post-condition</i>	Pengguna melihat info perubahan jadwal pelajaran

4.8.8 Skenario *Use Case* Lihat Materi Pelajaran

Kebutuhan selanjutnya adalah Materi Pelajaran yang berfungsi bagi siswa dan pengajar sebagai wadah untuk membaca *ebook* materi bahan pelajaran. Berikut *use case* scenario Materi Pelajaran yang dapat dilihat pada Tabel 4.18.

Tabel 4.18 Skenario *Use Case* Lihat Materi Pelajaran

<i>Objective</i>	Pengguna siswa maupun pengajar memilih menu Materi Pelajaran untuk menampilkan kelas dan keminatan, kemudian memilih jadwal pelajaran untuk menampilkan <i>ebook</i> sesuai pilihan.
<i>Actor</i>	Siswa, Pengajar
<i>Pre-condition</i>	Pengguna memilih menu Materi Pelajaran
<i>Main flow</i>	<ol style="list-style-type: none"> 1. Pengguna memilih menu materi pelajaran 2. Aplikasi akan menampilkan pilihan kelas 3. Aplikasi menampilkan pilihan keminatan 4. Aplikasi menampilkan pilihan mata pelajaran pada <i>action bar</i> 5. <i>Main layout</i> menampilkan <i>ebook</i> materi pelajaran
<i>Alternative flows</i>	Jika tidak ada pilihan mata pelajaran, terdapat pilihan cari lain untuk mencari <i>ebook</i> pada situs resmi Kemdikbud.
<i>Post-condition</i>	Pengguna melihat Materi Pelajaran yang dibagi berdasar kategori.

4.8.9 Skenario *Use Case* Lihat Aspirasi

Kebutuhan Lihat Aspirasi digunakan untuk melihat aspirasi pengguna untuk sekolah, baik kritik maupun saran serta penilaian yang membangun untuk sekolah. Berikut skenario *use case* lihat aspirasi dapat dilihat pada tabel 4.19.

Tabel 4.19 Skenario *Use Case* Lihat Aspirasi

<i>Objective</i>	Pengguna memilih menu Aspirasi untuk melihat daftar aspirasi dari pengguna
<i>Actor</i>	Siswa, Pengajar
<i>Pre-condition</i>	Pengguna memilih menu Aspirasi
<i>Main flow</i>	<ol style="list-style-type: none"> 1. Pengguna memilih menu Aspirasi 2. Aplikasi akan menampilkan daftar aspirasi
<i>Alternative flows</i>	Jika tidak terdapat koneksi internet, maka aplikasi tidak dapat menampilkan daftar aspirasi

<i>Post-condition</i>	Pengguna melihat daftar aspirasi
-----------------------	----------------------------------

4.8.10 Skenario *Use Case* Kirim Aspirasi

Kebutuhan Kirim Aspirasi digunakan untuk mengirim aspirasi pengguna untuk sekolah, baik kritik maupun saran serta penilaian yang membangun untuk sekolah. Berikut skenario *use case* lihat aspirasi dapat dilihat pada Tabel 4.20.

Tabel 4.20 Skenario *Use Case* Kirim Aspirasi

<i>Objective</i>	Pengguna memiliki menu Aspirasi
<i>Actor</i>	Siswa, Pengajar
<i>Pre-condition</i>	Pengguna memilih menu Aspirasi
<i>Main flow</i>	<ol style="list-style-type: none"> 1. Pengguna memilih menu Aspirasi 2. Aplikasi akan menampilkan pilihan sasaran aspirasi dan <i>form</i> aspirasi 3. Aplikasi menampilkan daftar aspirasi
<i>Alternative flows</i>	<ol style="list-style-type: none"> 2.1 Jika masukan data masih kosong, sistem akan memunculkan pesan notifikasi "Masukkan Aspirasi Terlebih Dahulu" 2.2 Jika tidak terdapat koneksi internet, maka aplikasi tidak dapat menyimpan aspirasi pada database dan menampilkannya pada daftar aspirasi
<i>Post-condition</i>	Pengguna melihat <i>form</i> aspirasi yang siap untuk diisi dan daftar aspirasi

4.8.11 Skenario *Use Case* Lihat Tugas

Use Case skenario Lihat Tugas digunakan bagi para siswa untuk melihat daftar tugas yang diberikan oleh tim pengajar yang sebelumnya telah ditambahkan. Berikut skenario *use case* Lihat Tugas yang dapat dilihat pada Tabel 4.21.

Tabel 4.21 Skenario *Use Case* Lihat Tugas

<i>Objective</i>	Pengguna memiliki menu Tugas untuk melihat daftar tugas dari tim pengajar.
<i>Actor</i>	Siswa, Pengajar
<i>Pre-condition</i>	Pengguna memilih menu Tugas
<i>Main flow</i>	<ol style="list-style-type: none"> 1. Pengguna memilih menu Tugas 2. Aplikasi akan menampilkan daftar tugas dengan detail index tugas, kelas, deadline.

<i>Alternative flows</i>	Jika tidak terdapat koneksi internet, aplikasi tidak dapat menampilkan daftar tugas
<i>Post-condition</i>	Pengguna dapat melihat daftar tugas

4.8.12 Skenario *Use Case* Unggah Tugas

Use Case skenario Unggah Tugas digunakan bagi para siswa untuk melihat melakukan aktivitas unggah. Berikut skenario *use case* Unggah Tugas yang dapat dilihat pada Tabel 4.22.

Tabel 4.22 Skenario *Use Case* Unggah Tugas

<i>Objective</i>	Pengguna memiliki menu Tugas untuk melakukan aktivitas unggah tugas
<i>Actor</i>	Siswa
<i>Pre-condition</i>	Pengguna memilih menu Tugas
<i>Main flow</i>	<ol style="list-style-type: none"> 1. Pengguna memilih menu Tugas 2. Pengguna menekan tombol <i>floating-button</i> pada halaman 3. Pengguna memasukkan nama teks berkas sesuai format 4. Pengguna menekan tombol unggah
<i>Alternative flows</i>	Jika tidak terdapat koneksi internet, aplikasi tidak dapat menampilkan daftar tugas
<i>Post-condition</i>	Berkas tugas tersimpan dalam <i>database</i> dan siap dilihat dan diunduh oleh pengajar

4.8.13 Skenario *Use Case* Tambah Tugas

Use Case skenario Tambah Tugas digunakan tim pengajar untuk menambah daftar tugas. Berikut skenario *use case* Tambah Tugas yang dapat dilihat pada Tabel 4.23.

Tabel 4.23 Skenario *Use Case* Tambah Tugas

<i>Objective</i>	Pengguna memiliki menu Tugas untuk melakukan aktivitas tambah daftar tugas
<i>Actor</i>	Pengajar
<i>Pre-condition</i>	Pengguna memilih menu Tugas
<i>Main flow</i>	<ol style="list-style-type: none"> 1. Pengguna memilih menu Tugas 2. Pengguna melakukan pengisian seluruh form tugas yang ada sesuai dengan ketentuan

	3. Pengguna menekan tombol tambah
<i>Alternative flows</i>	2.1 Jika tidak terdapat koneksi internet, aplikasi tidak dapat menambah dan menampilkan daftar tugas 2.2 Jika terdapat form yang kosong, aplikasi akan menampilkan pesan peringatan untuk melengkapi form yang tersedia
<i>Post-condition</i>	Tugas baru telah disimpan dan ditampilkan pada daftar tugas

4.8.14 Skenario *Use Case* Lihat Berkas Tugas

Use Case skenario Lihat Berkas Tugas digunakan tim pengajar untuk melihat berkas tugas yang diunggah oleh siswa. Berikut skenario *use case* Lihat Berkas Tugas yang dapat dilihat pada Tabel 4.24.

Tabel 4.24 Skenario *Use Case* Lihat Berkas Tugas

<i>Objective</i>	Pengguna memiliki menu Tugas untuk melakukan aktivitas tambah daftar tugas
<i>Actor</i>	Pengajar
<i>Pre-condition</i>	Pengguna memilih menu Tugas
<i>Main flow</i>	1. Pengguna memilih menu Tugas 2. Pengguna memilih tombol lihat berkas 3. Aplikasi akan menampilkan daftar berkas tugas yang telah diunggah siswa sebelumnya
<i>Alternative flows</i>	Jika tidak terdapat koneksi internet, aplikasi tidak dapat menampilkan daftar berkas tugas
<i>Post-condition</i>	Daftar tugas dari siswa ditampilkan pada laman aplikasi

4.8.15 Skenario *Use Case* Unduh Berkas Tugas

Use Case skenario Unduh Berkas Tugas digunakan tim pengajar untuk melakukan aktivitas unduh berkas tugas yang diunggah oleh siswa. Berikut skenario *use case* Unduh Berkas Tugas yang dapat dilihat pada Tabel 4.25.

Tabel 4.25 Skenario *Use Case* Unduh Berkas Tugas

<i>Objective</i>	Pengguna memiliki menu Tugas untuk melakukan aktivitas tambah daftar tugas
<i>Actor</i>	Pengajar
<i>Pre-condition</i>	Pengguna memilih menu Tugas

<i>Main flow</i>	<ol style="list-style-type: none"> 1. Pengguna memilih menu Tugas 2. Pengguna memilih tombol lihat berkas 3. Aplikasi akan menampilkan daftar berkas tugas yang telah diunggah siswa sebelumnya 4. Pengguna memilih salah satu atau beberapa daftar yang ada untuk diunduh berkasnya
<i>Alternative flows</i>	Jika tidak terdapat koneksi internet, aplikasi tidak dapat melakukan aktivitas unduh terhadap berkas tugas
<i>Post-condition</i>	Berkas tugas selesai diunduh dan tersimpan pada memori penyimpanan

4.8.16 Skenario Use Case Lihat Detil Akun

Kebutuhan fungsional Lihat Akun ini digunakan oleh pengguna siswa melihat data pengguna. Skenario *use case* untuk Lihat Detil Akun keluar ditunjukkan dalam Tabel 4.26.

Tabel 4.26 Skenario Use Case Lihat Detil Akun

<i>Objective</i>	Pengguna dapat melihat detil data akun
<i>Actor</i>	Siswa
<i>Pre-condition</i>	Pengguna memilih menu Akun
<i>Main flow</i>	<ol style="list-style-type: none"> 1. Pengguna memilih menu akun 2. Pengguna menampilkan data akun 3. Aplikasi menampilkan detil akun pengguna
<i>Alternative flows</i>	Jika tidak terdapat koneksi internet, aplikasi tidak dapat menampilkan detil akun pengguna
<i>Post-condition</i>	Aplikasi menampilkan detil informasi akun pengguna

4.8.17 Skenario Use Case Ubah Detil Akun

Kebutuhan fungsional Lihat Akun ini digunakan oleh pengguna siswa mengubah data pengguna. Skenario *use case* untuk ubah detil akun ditunjukkan dalam Tabel 4.27.

Tabel 4.27 Skenario Use Case Ubah Detil Akun

<i>Objective</i>	Pengguna dapat mengubah data akun
<i>Actor</i>	Siswa
<i>Pre-condition</i>	Pengguna memilih menu Akun
<i>Main flow</i>	<ol style="list-style-type: none"> 1. Pengguna memilih menu akun 2. Pengguna memilih tombol ubah profil

	3. Aplikasi menampilkan form ubah profil 4. Pengguna memilih tombol simpan 5. Aplikasi menampilkan halaman akun
<i>Alternative flows</i>	Jika tidak terdapat koneksi internet, aplikasi tidak dapat menampilkan detil akun pengguna dan melakukan perubahan data
<i>Post-condition</i>	Aplikasi menyimpan informasi detil akun baru

4.8.18 Skenario Use Case Ubah Password

Kebutuhan fungsional Lihat Akun ini digunakan oleh pengguna siswa mengubah kata sandi pengguna. Skenario *use case* untuk ubah *password* ditunjukkan dalam Tabel 4.28.

Tabel 4.28 Skenario Use Case Ubah Detil Akun

<i>Objective</i>	Pengguna dapat mengubah kata sandi akun
<i>Actor</i>	Siswa
<i>Pre-condition</i>	Pengguna memilih menu Akun
<i>Main flow</i>	1. Pengguna memilih menu akun 2. Pengguna memilih tombol ubah profil 3. Pengguna memilih tombol ubah <i>password</i> 4. Pengguna memasukkan kata sandi baru 5. Pengguna memilih tombol ganti
<i>Alternative flows</i>	Jika tidak terdapat koneksi internet, aplikasi tidak dapat mengganti kata sandi pengguna
<i>Post-condition</i>	Aplikasi menyimpan informasi password baru

4.8.19 Skenario Use Case Pendaftaran

Kebutuhan fungsional ini digunakan oleh untuk mendaftarkan pengguna siswa baru pengguna aplikasi ini. Skenario *use case* untuk Pendaftaran ditunjukkan dalam Tabel 4.29.

Tabel 4.29 Skenario Use Case Pendaftaran

<i>Objective</i>	Pengguna dapat mendaftarkan akun baru untuk siswa
<i>Actor</i>	Pengajar
<i>Pre-condition</i>	Pengajar memilih menu pendaftaran
<i>Main flow</i>	1. Pengajar memilih menu pendaftaran 2. Aplikasi menampilkan pilihan akun siswa

	3. Aplikasi memilih tombol siswa 4. Aplikasi menampilkan form masukan data untuk mendaftarkan akun baru. 5. Aplikasi memilih tombol daftar
<i>Alternative flows</i>	Jika tidak terkoneksi dengan internet, sistem akan memunculkan pesan notifikasi “Pendaftaran gagal”
<i>Post-condition</i>	Pengajar berhasil mendaftarkan akun baru untuk siswa

4.8.20 Skenario Use Case Keluar

Kebutuhan fungsional terakhir yang terdapat dalam diagram *use case* yaitu *use case* keluar. Aktivitas keluar ini digunakan oleh pengguna ketika ingin keluar dari sistem dan mengakhiri aktivitas didalamnya. Skenario *use case* untuk aktivitas keluar ditunjukkan dalam Tabel 4.30.

Tabel 4.30 Skenario Use Case Keluar

<i>Objective</i>	Pengguna dapat keluar dari sistem
<i>Actor</i>	Pengajar
<i>Pre-condition</i>	Pengguna telah melakukan aktivitas masuk dan <i>valid</i> .
<i>Main flow</i>	1. Pengguna memilih pilihan keluar 2. Sistem melakukan <i>destroy session</i> dan menampilkan halaman login
<i>Alternative flows</i>	Jika tidak terdapat koneksi internet, sistem menolak untuk keluar sistem
<i>Post-condition</i>	Pengguna keluar dari sistem.

4.8.21 Skenario Use Case Keluar

Kebutuhan fungsional terakhir yang terdapat dalam diagram *use case* yaitu *use case* keluar. Aktivitas keluar ini digunakan oleh pengguna ketika ingin keluar dari sistem dan mengakhiri aktivitas didalamnya. Skenario *use case* untuk aktivitas keluar ditunjukkan dalam Tabel 4.31.

Tabel 4.31 Skenario Use Case Keluar

<i>Objective</i>	Pengguna dapat keluar dari sistem
<i>Actor</i>	Siswa
<i>Pre-condition</i>	Pengguna telah melakukan aktivitas masuk dan <i>valid</i> .
<i>Main flow</i>	1. Pengguna memilih menu akun 2. Pengguna memilih tombol keluar 3. Sistem melakukan <i>destroy session</i> dan menampilkan

	hamalan login
<i>Alternative flows</i>	Jika tidak terdapat koneksi internet, sisten menolak untuk keluar sistem
<i>Post-condition</i>	Pengguna keluar dari sistem.



BAB 5 PERANCANGAN DAN IMPLEMETASI

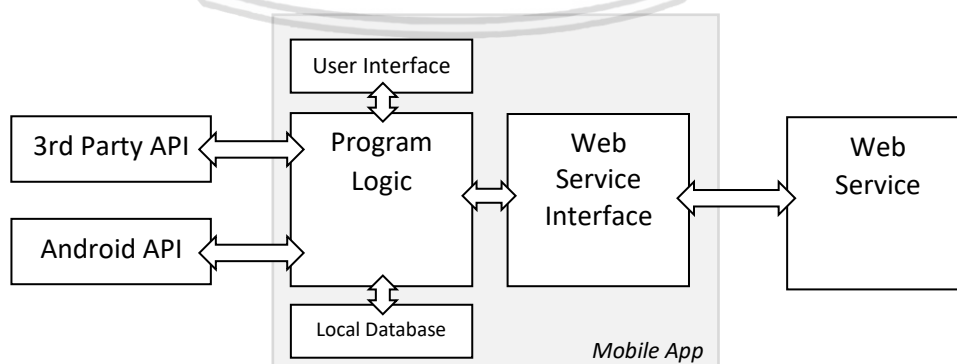
Pada bab ini membahas mengenai perancangan aplikasi pada *e-learning* menggunakan metode *User Centered Design* pada studi kasus SMAN 3 Sidoarjo. Tahapan yang dilakukan peneliti terdapat alur kerja aplikasi secara umum. Perancangan umum aplikasi menjelaskan proses kerja dari pengguna, basis data dan media yang digunakan untuk mengakses basis datanya. Tahap perancangan berikutnya adalah perancangan perangkat lunak, dalam tahap ini terdapat beberapa langkah yaitu perancangan arsitektur sistem, perancangan diagram arsitektur, perancangan basis data dan perancangan antarmuka aplikasi.

5.1 Perancangan Sistem

Pada tahap perancangan berdasarkan metode UCD berada pada tahap *Product design solution* dan *Evaluate design against user requirement* yang menerapkan perancangan antarmuka beserta iterasinya pada sub-bab 5.1.6. Tahap perancangan diawali dengan perancangan umum sistem yang menggambarkan alur kerja sistem secara umum. Perancangan umum sistem menjelaskan tentang alur proses kerja dari pengguna, basis data, dan media yang mengaksesnya. Selanjutnya adalah tahap perancangan arsitektur sistem, perancangan basis data, perancangan diagram *class*, perancangan diagram *sequence*, perancangan algoritme dan perancangan antarmuka pengguna.

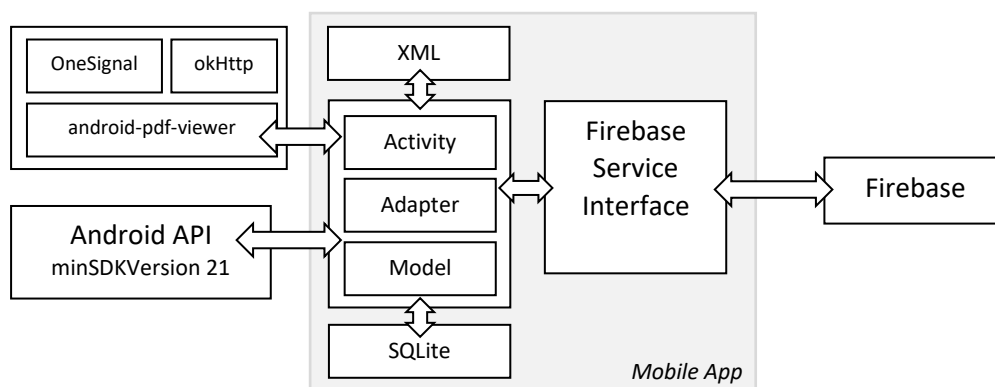
5.1.1 Perancangan Arsitektur Sistem

Perancangan arsitektur *e-learning* ini dibangun menggunakan arsitektur *front-end* dan *back-end* yang memisahkan antara pengolahan antarmuka sistem dan pengolahan data yang ada dalam arsitektur *back-end*. Konsep *mobile application* pada perangkat *mobile* dijalankan dengan sistem operasi Android. Konsep aplikasi ini dibangun menggunakan Bahasa Java serta XML untuk tampilan aplikasinya. Berikut ini merupakan rancangan arsitektur sistem *e-learning* berbasis android yang dapat dilihat pada gambar 5.1 dan 5.2.



Gambar 5.1 Arsitektur Sistem Aplikasi Mobile

Sumber: (STAPIĆ, 2013)



Gambar 5.2 Arsitektur Sistem Aplikasi E-Learning SMANTIGDA

Arsitektur sistem pada bagian *mobile app* terdapat empat bagian, *User Interface*, *Program Logic*, *Web Service Interface* dan *Local Database*. Pada aplikasi yang dikembangkan menggunakan empat bagian tersebut untuk:

1. *User Interface* : terdapat berkas XML untuk tampilan *activity* aplikasi
2. *Program Logic* : terdapat kumpulan *java class* yang terdiri dari *class activity*, *model* dan *adapter*.
3. *Local Database* : terdapat SQLite yang digunakan dalam aplikasi yang dikembangkan
4. *Web Service Interface* : terdapat *Firebase Service Interface* yang menjembantani antara *mobile app* dan *Firebase* itu sendiri.

Android API dan 3rd *Party* API juga digunakan dalam aplikasi yang sedang dikembangkan.

1. Android API menerapkan minSDKVersion 21 atau setara dengan OS Android *Lollipop* keatas.
2. 3rd *Party* API terdapat OneSignal, okHttp dan android-pdf-viewer
 - a. OneSignal : digunakan untuk fitur yang membutuhkan *push notification*
 - b. okHttp : digunakan untuk penanganan HTTP pada aplikasi
 - c. android-pdf-viewer : digunakan untuk menampilkan berkas *.pdf dalam aplikasi.

5.1.2 Perancangan Basis Data

Perancangan basis data akan menggunakan skema diagram karena basis data yang digunakan pada penelitian ini adalah *Firebase Real-time Database* yang berbasis noSQL, yang menggunakan *database* noSQL antara lain akun siswa,

aspirasi, info jadwal, *timeline* akademik, tugas siswa dan berkas tugas. Selain itu juga terdapat fitur latihan soal yang menggunakan penyimpanan data lokal dengan menggunakan SQLite, pengembang menggunakan penyimpanan data lokal dengan alasan data yang disimpan untuk akun pengajar dan latihan soal bersifat statis, data dapat saja berubah namun dalam jangka waktu lama.

5.1.2.1 Skema Basis Data Akun Siswa

Berikut adalah skema basis data akun siswa yang akan digunakan dalam penelitian ini. Terdapat atribut *idAkunSiswa*, *alamatSiswa*, *bulanLahir*, *namaSiswa*, *tahunLahir*, *tanggalLahir*, *tempatLahir* bertipe *string*. Skema basis data akun siswa ditunjukkan pada Gambar 5.3.

5.1.2.2 Skema Basis Data Aspirasi

Berikut adalah skema basis data Aspirasi yang akan digunakan dalam penelitian ini. Skema basis data aspirasi ditunjukkan pada Gambar 5.4.



Gambar 5.3 Perancangan Basis Data Akun Siswa

Gambar 5.4 Perancangan Basis Data Aspirasi

Keterangan dari skema basis data akun siswa dapat dilihat pada Tabel 5.1.

Tabel 5.1 Tabel Keterangan Skema Basis Data Akun Siswa

No	Nama Filed	Tipe Data	Keterangan
1	IdAkunSiswa	<i>String</i>	Untuk menyimpan id akun siswa
2	alamatSiswa	<i>String</i>	Untuk menyimpan data alaman siswa
3	namaSiswa	<i>String</i>	Untuk menyimpan data nama siswa

4	tanggalLahir	<i>String</i>	Untuk menyimpan data tanggal lahir siswa
5	tempatLahir	<i>String</i>	Untuk menyimpan data tempat lahir siswa

Keterangan dari skema basis data aspirasi dapat dilihat pada Tabel 5.2.

Tabel 5.2 Tabel Keterangan Skema Basis Data Aspirasi

No	Nama Filed	Tipe Data	Keterangan
1	Aspirasi	<i>String</i>	Untuk menyimpan data aspirasi pengguna
2	aspirasiId	<i>String</i>	Untuk menyimpan id aspirasi pengguna
3	sasaranAspirasi	<i>String</i>	Untuk menyimpan sasaran aspirasi pengguna

5.1.2.3 Skema Basis Data Info Jadwal

Berikut adalah skema basis data info jadwal yang akan digunakan dalam penelitian ini. Skema basis data info jadwal ditunjukkan pada Gambar 5.5.

```

"Info Jadwal" : {
  "type": "object"
  "properties":{
    "id":{
      "type": "object"
      "properties":
      } "infoJadwal":{
        "type":"string"
      }, "tglInfo":{
        "type":"string"
      },
    },
  },
}

```

Gambar 5.5 Perancangan Basis Data Info Jadwal

Keterangan dari skema basis data info jadwal dapat dilihat pada Tabel 5.3.

Tabel 5.3 Tabel Keterangan Skema Basis Data Aspirasi

No	Nama Filed	Tipe Data	Keterangan
1	infoJadwal	<i>String</i>	Untuk menyimpan data info perubahan jadwal
2	tglInfo	<i>String</i>	Untuk menyimpan data tanggal perubahan jadwal

5.1.2.4 Skema Basis Data Timeline Akademik

Berikut adalah skema basis data timeline akademik yang akan digunakan dalam penelitian ini. Skema basis data timeline akademik ditunjukkan pada Gambar 5.6.

```

"Timeline Akademik" : {
  "type": "object"
  "properties":{
    "id":{
      "type": "object"
      "properties":{
        "judul":{
          "type":"string"
        }, "tanggal":{
          "type":"string"
        }, "tlid":{
          "type":"string"
        },
      },
    },
  },
}

```

Gambar 5.6 Perancangan Basis Data *Timeline Akademik*

Keterangan skema basis data timeline akademik dapat dilihat pada Tabel 5.4.

Tabel 5.4 Tabel Keterangan Skema Basis Data Timeline Akademik

No	Nama Filed	Tipe Data	Keterangan
1	judul	<i>String</i>	Untuk menyimpan data judul <i>timeline</i> kegiatan akademik
2	tanggal	<i>String</i>	Untuk menyimpan data tanggal kegiatan akademik
3	tlId	<i>String</i>	Untuk menyimpan data id kegiatan akademik

5.1.2.5 Skema Basis Data Berkas Tugas

Berikut adalah skema basis data Uploads yang akan digunakan dalam penelitian ini. Dalam basis data ini terdapat atribut nama bertipe sting, url bertipe string dan idAkunSiswa bertipe string. Skema basis data uploads ditunjukkan pada Gambar 5.7.

```

"uploads" : {
  "type": "object"
  "properties": {
    "id" : {
      "type": "object"
      "properties": {
        "name": {
          "type": "string"
        }, "url": {
          "type": "string"
        }, "idAkunSiswa": {
          "type": "string"
        }, , , ,
      }
    }
  }
}

```

Gambar 5.7 Perancangan Basis Data Berkas Tugas

Keterangan dari skema basis data berkas tugas dapat dilihat pada Tabel 5.5.

Tabel 5.5 Tabel Keterangan Skema Basis Data Berkas Tugas

No	Nama Filed	Tipe Data	Keterangan
1	name	<i>String</i>	Untuk menyimpan data nama berkas tugas
2	url	<i>String</i>	Untuk menyimpan data url berkas dari <i>database</i>
3	idAkunSIswa	<i>String</i>	Untuk menyimpan id akun siswa

5.1.2.6 Skema Basis Data Tugas Siswa

Berikut adalah skema basis data tugas siswa yang akan digunakan dalam penelitian ini. Pada skema basis data tugas siswa ini terdapat atribut bulanTugas bertipe string, detil tugas bertipe string, idTugas bertipe string, indexTugas bertipe string, kelasTugas bertipe string, jamTugas bertipe string, kelasTugas bertipe string, matpelTugas bertipe string, meridiem bertipe string, tahunTugas bertipe string, temaTugas bertipe string dan tanggalTuags bertipe string. Skema basis data tugas siswa ditunjukkan pada Gambar 5.8.

```

"tugas siswa" : {
  "type" : "object"
  "properties":{
    "id":{
      "type":"object"
      "properties":{
        "detilTugas":{
          "type":"string"
        }, "idTugas":{
          "type":"string"
        }, "indexTugas":{
          "type":"string"
        }, "jamTugas":{
          "type":"string"
        }, "kelasTugas":{
          "type":"string"
        }, "matpelTugas":{
          "type":"string"
        }, "temaTugas":{
          "type":"string"
        }, "tanggalTugas":{
          "type":"string"
        }
      },
    },
  },
}

```

Gambar 5.8 Perancangan Basis Data Tugas Siswa

Keterangan dari skema basis data tugas siswa dapat dilihat pada Tabel 5.6.

Tabel 5.6 Tabel Keterangan Skema Basis Data Tugas Siswa

No	Nama Filed	Tipe Data	Keterangan
1	detilTugas	<i>String</i>	Untuk menyimpan data detil tugas
2	idTugas	<i>String</i>	Untuk menyimpan data id tugas
3	indexTugas	<i>String</i>	Untuk menyimpan data index tugas
4	jamTugas	<i>String</i>	Untuk menyimpan data jam tugas
5	kelasTugas	<i>String</i>	Untuk menyimpan data kelas
6	matpelTugas	<i>String</i>	Untuk menyimpan data judul mata pelajaran

5.1.2.7 Skema Basis Data Latihan Soal

Berikut adalah skema basis data latihan soal yang akan digunakan pada penelitian ini. Basis data pada fitur ini juga menggunakan basis data sql karena sifat data yang statis dan jumlah yang banyak menjadi alasan mengapa peneliti memilih menggunakan basis data sql. ER Diagram basis data latihan soal dapat dilihat pada Gambar 5.9.

LatihanSoal

Id
question
choice1
choice2
choice3
choice4
answer

Gambar 5.9 Perancangan Basis Data ER Diagram Latihan Soal

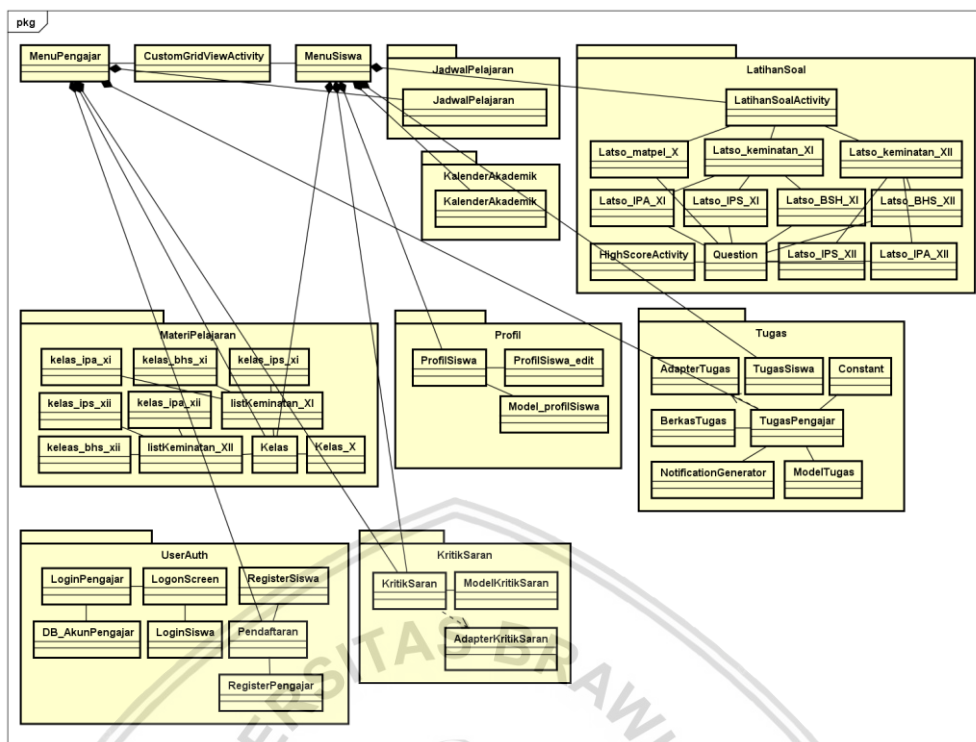
Keterangan dari skema basis data tugas siswa dapat dilihat pada Tabel 5.7

Tabel 5.7 Tabel Keterangan Skema Basis Data Tugas Siswa

No	Nama Filed	Tipe Data	Keterangan
1	Id	Integer	Nomor pengenal yang dimiliki tabel latihan soal
2	question	TEXT	Kolom pertanyaan untuk menyimpan pertanyaan latihan soal
3	choice1	TEXT	Kolom pilihan jawaban pertama untuk menyimpan opsi jawaban
4	choice2	TEXT	Kolom pilihan jawaban kedua untuk menyimpan opsi jawaban
5	choice3	TEXT	Kolom pilihan jawaban kedua untuk menyimpan opsi jawaban
6	choice4	TEXT	Kolom pilihan jawaban kedua untuk menyimpan opsi jawaban
7	answer	TEXT	Kolom jawaban untuk menyimpan jawaban

5.1.3 Perancangan *Class Diagram*

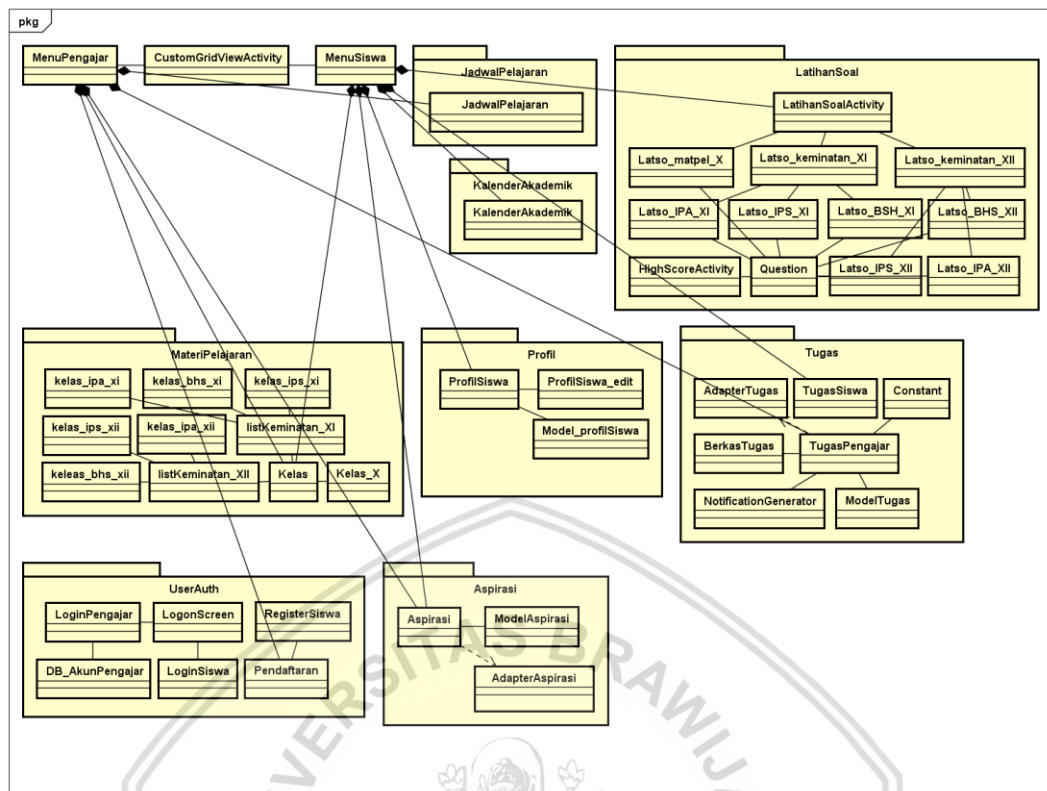
Perancangan *class diagram* merupakan diagram yang digunakan untuk menunjukkan kelas dan relasinya dalam suatu sistem. Kelas yang digunakan akan dikelompokkan menjadi beberapa paket berdasarkan fungsi-fungsinya. Untuk perancangan kelas diagram ditunjukkan pada Gambar 5.10.



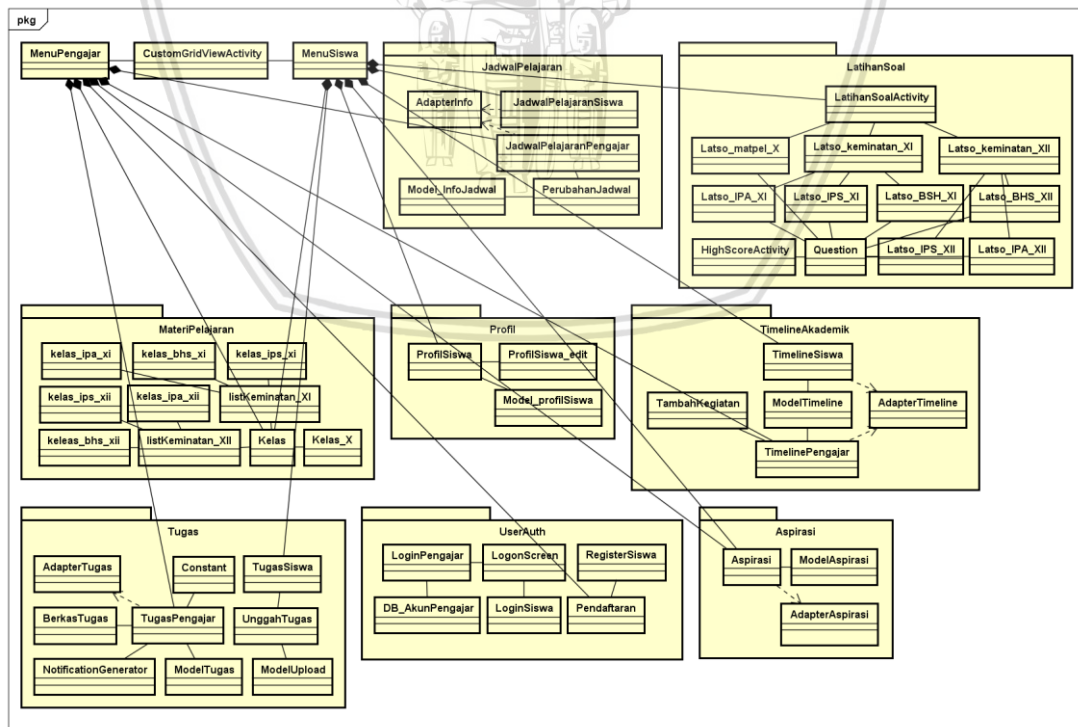
Gambar 5.10 Perancangan *Class Diagram*

Mengacu pada perubahan yang ada, *class diagram* juga terdapat perubahan pada beberapa *class*-nya. Pada gambar 5.11 menunjukkan perubahan *class diagram* pada iterasi 1 dengan perubahan dihapusnya *class* pendaftaran untuk pengajar, namun tetap ada pendaftaran bagi siswa namun hanya dapat dilakukan oleh sisi pengajar saja. Serta perubahan nama fitur juga terjadi yaitu KritikSaran menjadi Aspirasi.

Terdapat perubahan lagi pada *class diagram* yang ditunjukkan pada gambar 5.12. Perubahan yang terjadi yaitu adanya perubahan nama fitur, dari KalenderAkademik menjadi TimelineAkademik, serta terdapat sub-fitur baru yaitu tambah kegiatan pada TimelineAkademik dan tambah info perubahan jadwal pada JadwalPelajaran.



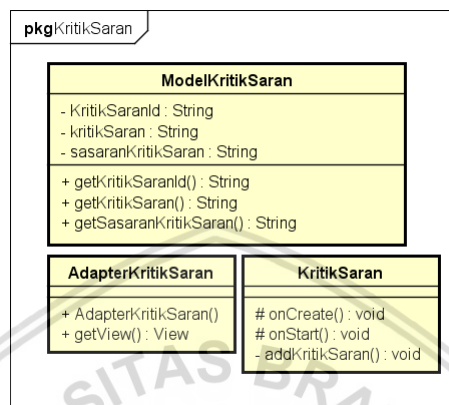
Gambar 5.11 Perancangan Class Diagram (Iterasi 1)



Gambar 5.12 Perancangan Class Diagram (Iterasi 2)

5.1.3.1 Perancangan Class Diagram Paket KritikSaran

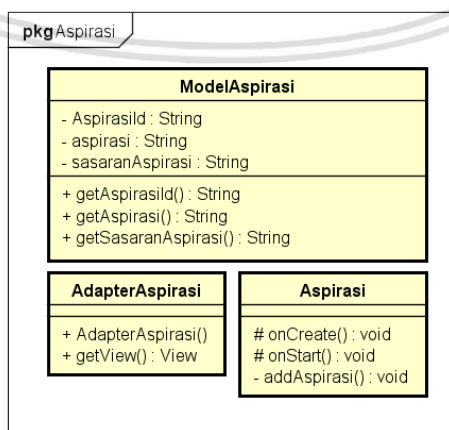
Paket KritikSaran ini terdapat tiga kelas yaitu ModelKritikSaran, AdapterKritikSaran dan KritikSaran itu sendiri sebagai *main class*-nya. Paket KritikSaran ini digunakan pada fitur Kritik Saran pada aplikasi, baik dari sisi Pengajar ataupun dari sisi Siswa. Perancangan klas diagram paket KritikSaran dapat dilihat pada Gambar 5.13.



Gambar 5.13 Perancangan Class Diagram Paket KritikSaran

5.1.3.2 Perancangan Class Diagram Paket Aspirasi (Iterasi 1)

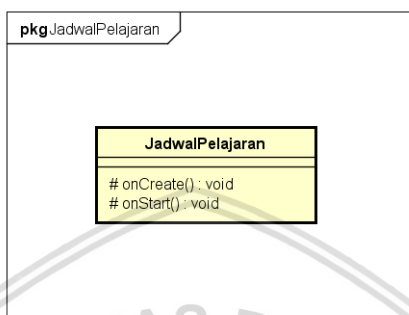
Paket ini merupakan pembaharuan dari paket sebelumnya yaitu KritikSaran. Paket Aspirasi terdapat tiga kelas yaitu klas Aspirasi, AdapterAspirasi, dan ModelAspirasi. Pada klas Aspirasi terdapat atribut-atribut untuk menampilkan seluruh data yang digunakan pada menu tersebut. Pada klas ModelAspirasi terdapat atribut dan metode untuk melakukan kegiatan *setter-getter*, dan klas AdapterAspirasi untuk merubah data dari *database* ke dalam bentuk *array-list* sebelum ditampilkan pada klas Aspirasi. *Diagram Class* Paket Aspirasi dapat dilihat pada Gambar 5.14.



Gambar 5.14 Perancangan Class Diagram Paket Aspriasi (Iterasi 1)

5.1.3.3 Perancangan Class Diagram Paket JadwalPelajaran

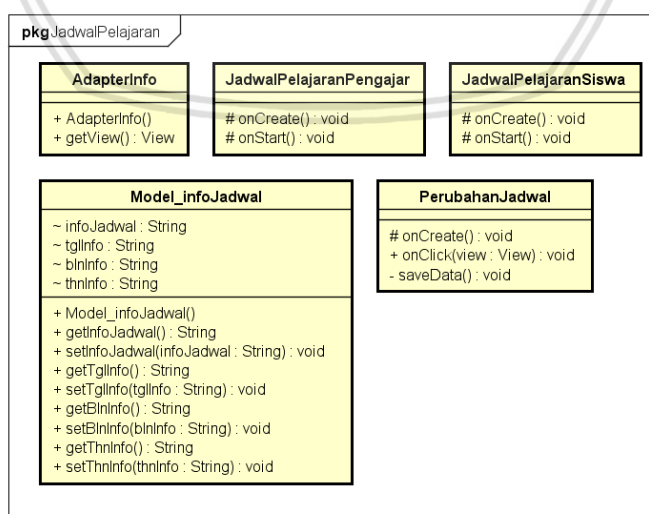
Paket jadwal pelajaran hanya terdapat satu klas saja yaitu klas JadwalPelajaran. Pada klas tersebut memiliki atribut *toolbar* dan *spinner* yang digunakan untuk pemilihan klas. Dan pada klas ini terdapat satu buah *method* *onCreate*, kedua atributnya dieksekusi dan dijalankan di *method* *onCreate* tersebut. Perancangan klas diagram paket JadwalPelajaran dapat dilihat pada Gambar 5.15.



Gambar 5.15 Perancangan Diagram Class Package JadwalPelajaran

5.1.3.4 Perancangan Class Diagram Paket JadwalPelajaran (Iterasi 1)

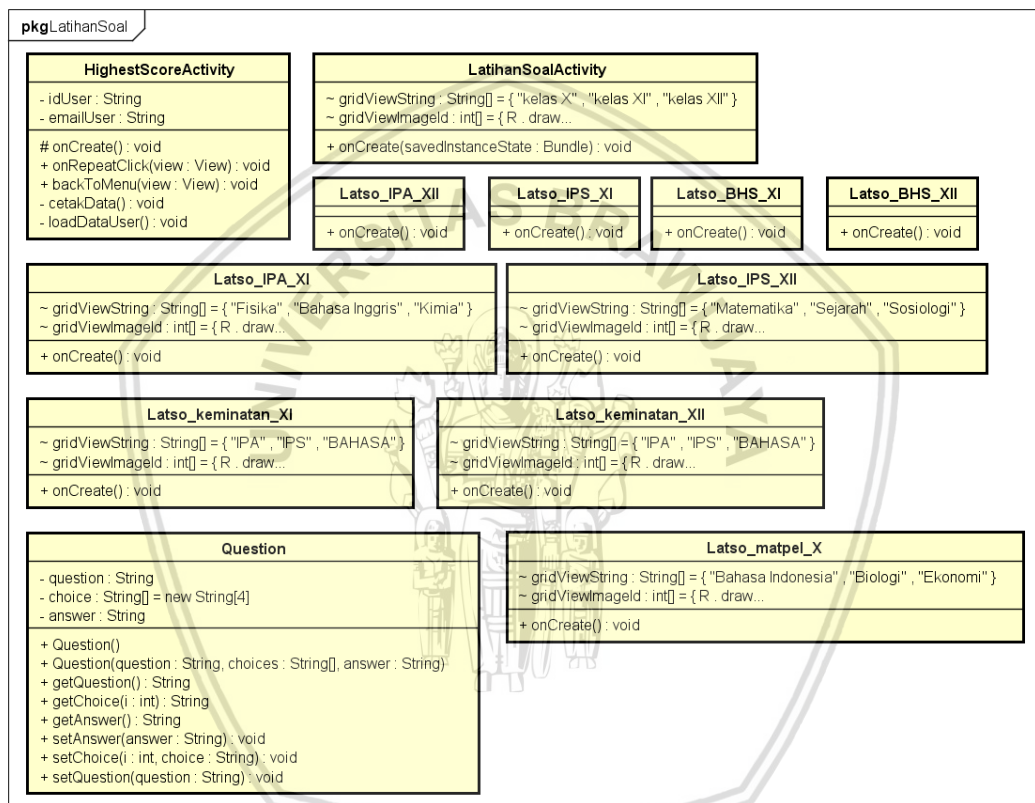
Adanya perubahan kebutuhan yang terjadi pada paket klas JadwalPelajaran dari sebelumnya. Paket JadwalPelajaran kali ini terdapat lima buah *class*, JadwalPelajaranPengajar dan JadwalPelajaranSiswa merupakan klas utama dari fitur jadwal pelajaran dari sisi pengajar dan siswa. Kemudian juga terdapat class PerubahanJadwal sebagai *main class* dari fitur tambah info perubahan jadwal, seluruh data dimodelkan pada *class* model dengan atribut *infoJadwal*, *tglInfo*, *blnInfo* dan *thnInfo*. Diagram Class paket JadwalPelajaran dapat dilihat pada Gambar 5.16.



Gambar 5.16 Perancangan Class Diagram Paket Jadwal Pelajaran (Iterasi 1)

5.1.3.5 Perancangan Class Diagram Paket LatihanSoal

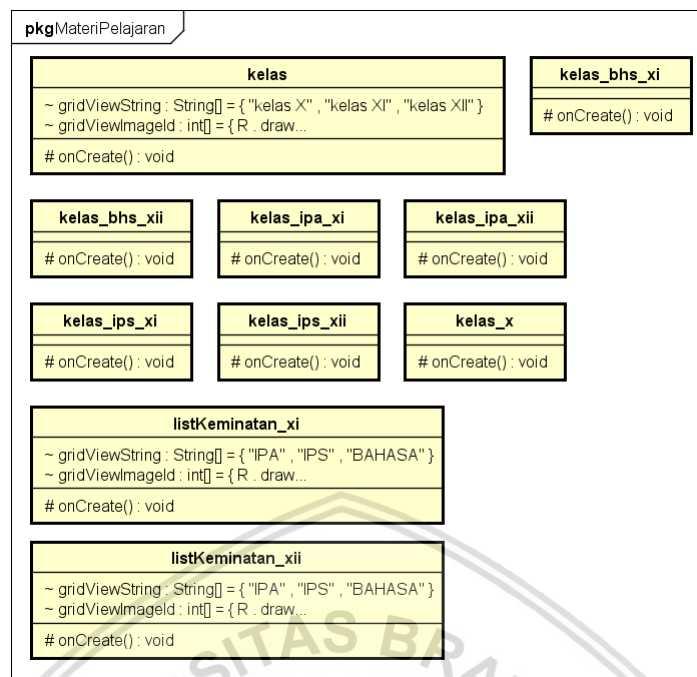
Paket LatihanSoal terdapat dua belas klas, paket ini terdapat klas LatihanSoalActivity yang diakses pertama kali dari menu utama, klas tersebut menampilkan pilihan klas untuk latihan soal, kemudian akan diarahkan kembali menuju ke arah yang lebih spesifik hingga dapat memilih latihan soal. Terdapat klas Question dimana klas tersebut merupakan *activity* dari fitur latihan soal, dan pada akhir latihan soal akan menuju ke *class* HighestScoreActivity untuk menampilkan hasil dari latihan soal tersebut. Paket perancangan klas diagram Latihan Soal dapat dilihat pada Gambar 5.17.



Gambar 5.17 Perancangan Class Diagram Paket LatihanSoal

5.1.3.6 Perancangan Class Diagram Paket MateriPelajaran

Paket perancangan klas diagram MateriPelajaran ini memiliki sepuluh buah klas dengan *class* kelas yang menampilkan pilihan kelas, kemudian pengguna akan diarahkan pada pilihan yang makin spesifik untuk melihat materi pelajaran. Perancangan klas diagram paket MateriPelajaran dapat dilihat pada Gambar 5.18.



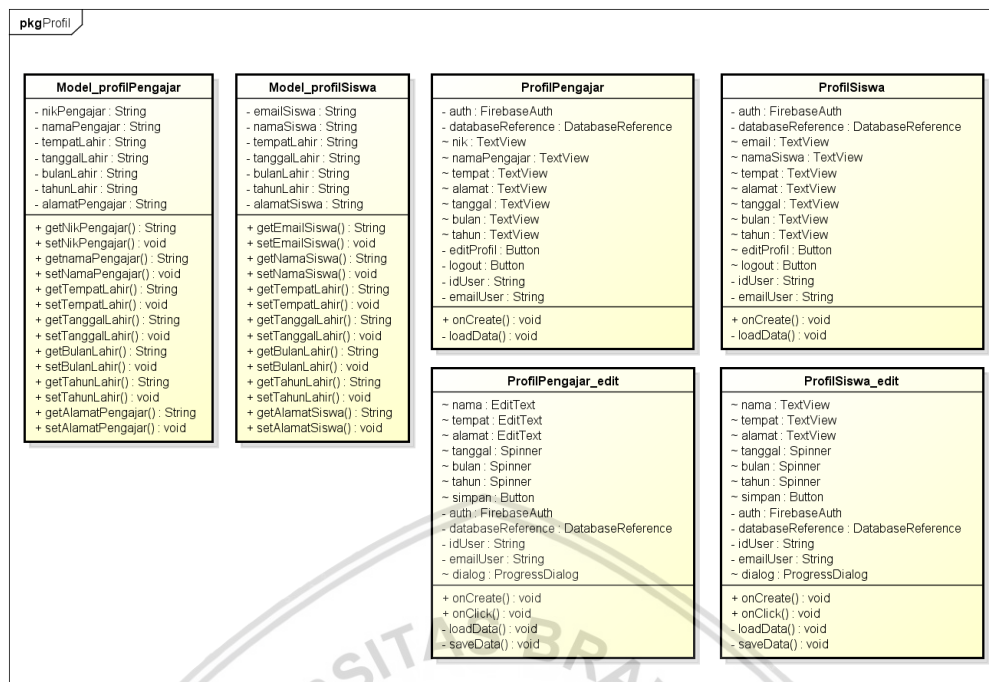
Gambar 5.18 Perancangan Class Diagram Paket Materi Pelajaran

5.1.3.7 Perancangan Class Diagram Paket Profil

Paket profil terdapat enam klas, dua klas ProfilPengajar dan ProfilSiswa sebagai klas utama dari menu profil, kemudian dua klas ProfilPengajar_edit dan kelas ProfilSiswa_edit untuk menu ubah profil, dan dua klas model dari masing-masing sisi pengguna. Pada klas Profil terdapat atribut sesuai apa yang ditampilkan pada halaman profil yaitu *email*, nama, tempat tanggal lahir, dan alamat. Kemudian juga terdapat atribut FirebaseAuth untuk menghubungkan data dengan *database* Firebase.

Pada klas ProfilPengajar_edit dan ProfilSiswa_edit terdapat atribut-atribut yang sama dengan atribut pada klas ProfilPengajar dan ProfilSiwa, namun berbeda cara eksekusinya, pada klas ProfilPengajar dan ProfilSiswa dieksekusi dengan menggunakan metode loadData() untuk menampilkan data dari *database*, namun untuk klas ProfilPengajar_edit dan ProfilSiswa_edit dieksekusi dengan menggunakan metode setData() untuk mengambil data dari halaman profilPegajar dan ProfilSiswa, kemudian terdapat metode saveData() untuk menyimpan data pada database.

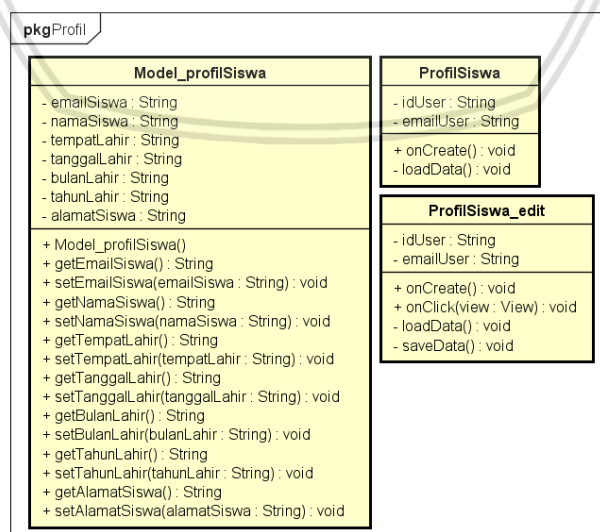
Dua *method* terakhir yaitu model_profilPengajar dan model_profilSiswa merupakan metode *setter-getter* dari klas ProfilPengajar, ProfilSiswa, ProfilPengajar_edit dan ProfilSiswa_edit. Paket klas diagram dapat dilihat pada Gambar 5.19.



Gambar 5.19 Perancangan Class Diagram Paket Profil

5.1.3.8 Perancangan Class Diagram Paket Profil (Iterasi 1)

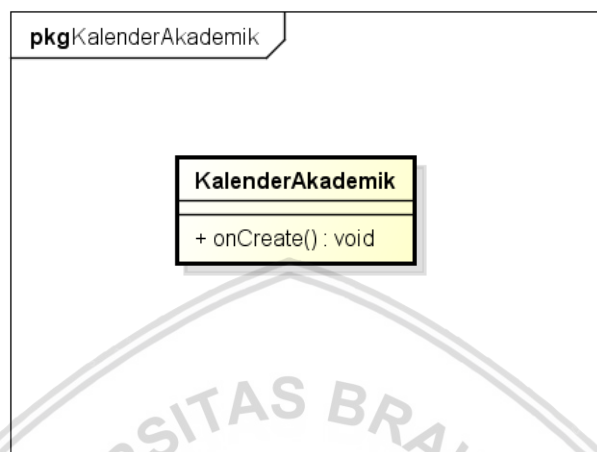
Paket kelas Profil setelah terjadinya barubahan ini terdapat tiga buah klas dengan Profilsiswa sebagai *main class*-nya. Pengguna dapat melihat profil siswa seperti yang ada pada atribut kelas Model_profilSiswa yaitu emailSiswa, namaSiswa, tempatLahir, tanggalLahir, bulanLahir, tahunLahir dan alamatSiswa. Seluruh info profil tersebut dapat diubah dengan class ProfilSiswa_edit. Perancangan klas diagram paket Profil dapat dilihat pada Gambar 5.20.



Gambar 5.20 Perancangan Class Paket Profil (Iterasi 1)

5.1.3.9 Perancangan Class Diagram Paket KalenderAkademik

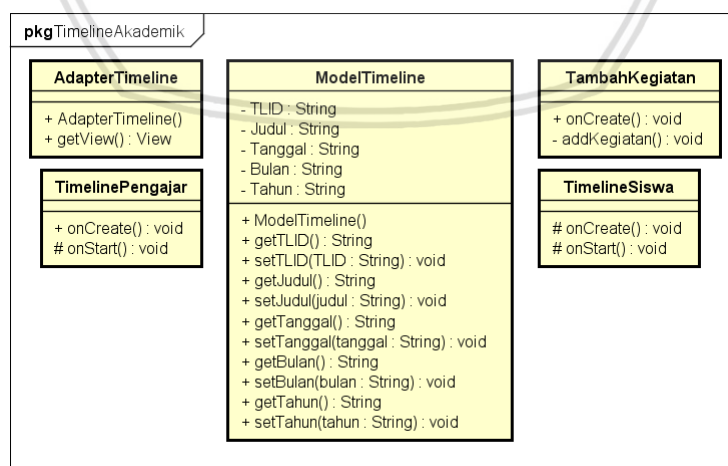
Paket KalenderAkademik ini hanya terdapat satu klas saja, dan hanya terdapat *method* onCreate(), karena pada klas ini hanya menampilkan data berbentuk *timeline* jawal kegiatan. Perancangan klas diagram paket KalenderAkademik ini dapat dilihat pada Gambar 5.21.



Gambar 5.21 Perancangan Class Diagram Paket KalenderAkademik

5.1.3.10 Perancangan Class Diagram Paket TimelineAkademik (Iterasi 1)

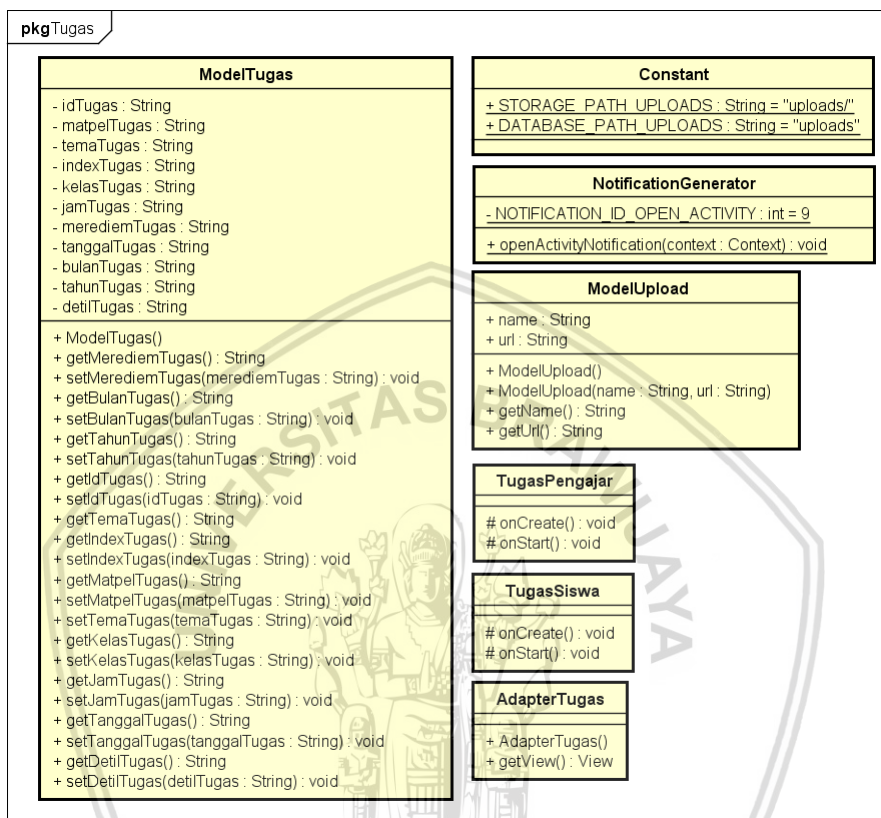
Paket TimelineAkademik setelah mengalami perubahan ini memiliki lima buah klas dengan TimelineSiswa dan TimelinePengajar sebagai *main class*-nya, terdapat klas TambahKegiatan yang dapat digunakan pengguna pengajar untuk menambah daftar kegiatan akademik. Kegiatan akademik tersebut menyimpan data seperti yang ada pada atribut kelas ModelTimeline yaitu Judul, Tanggal, Bulan, dan Tahun. Perancangan klas diagram paket TimelineAkademik dapat dilihat pada Gambar 5.22.



Gambar 5.22 Perancangan Class Diagram Paket TimelineAkademik (Iterasi 1)

5.1.3.11 Perancangan Class Diagram Paket Tugas

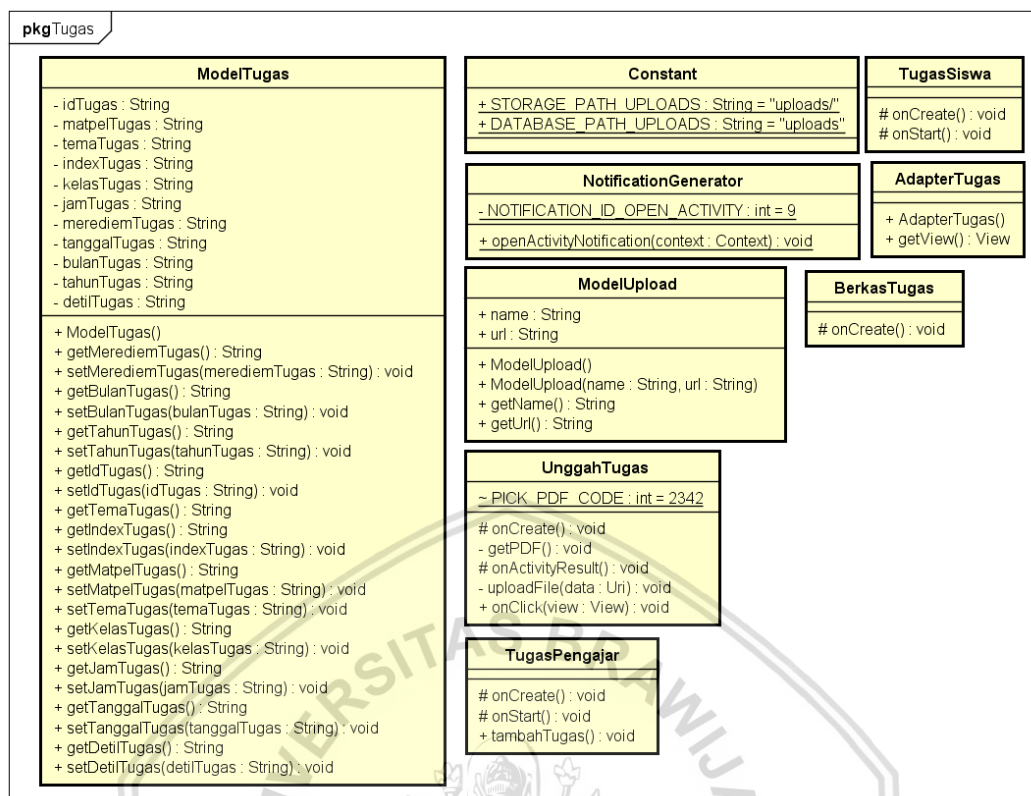
Paket Tugas ini berisi kelas-kelas yang digunakan dalam fitur tugas dari kedua sisi pengguna, dari sisi siswa menerima info tugas dari sisi pengajar dengan main *class* TugasPengajar dan TugasSiswa. Perancangan klas diagram paket Tugas dapat dilihat pada Gambar 5.23.



Gambar 5.23 Perancangan Class Diagram Paket Tugas

5.1.3.12 Perancangan Class Diagram Paket Tugas (Iterasi 1)

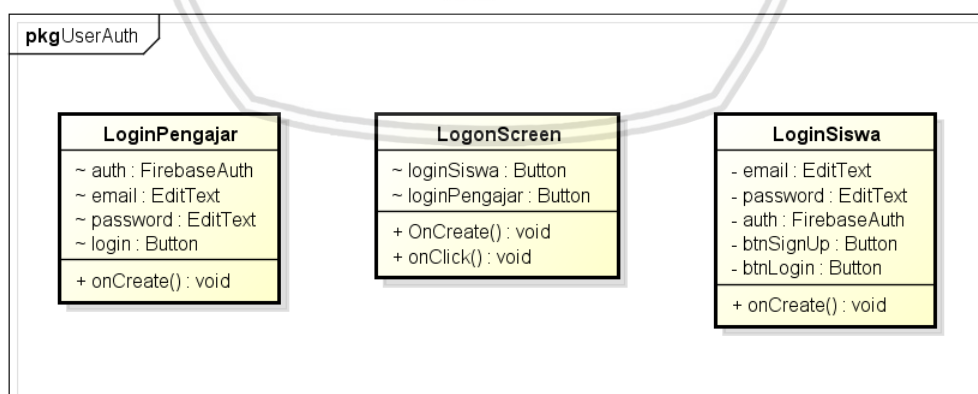
Paket Tugas ini setelah mengalami perubahan terdapat sembilan buat klas antara lain klas ModelTugas sebagai klas model, klas Constant sebagai klas dimana menginisialisasi *path* dari data yang dikirim ke *database*, klas NotificationGenerator sebagai klas untuk setting notifikasinya, klas ModelUpload untuk klas model dari fitur unggah, klas UnggahTugas untuk klas fitur unggah tugas, klas TugasPengajar sebagai *main class* dari fitur tugas sisi pengajar, klas TugasSiswa sebagai *main class* dari fitur tugas sisi siswa, klas AdapterTugas digunakan sebagai klas adapter dari data daftar tugas kemudian yang terakhir terdapat klas BerkasTugas yang digunakan untuk pengajar melihat daftar berkas tugas dari siswa. Perancangan klas diagram paket Tugas dapat dilihat pada Gambar 5.24.



Gambar 5.24 Perancangan Class Paket Tugas (Iterasi 1)

5.1.3.13 Perancangan Class Diagram Paket UserAuth

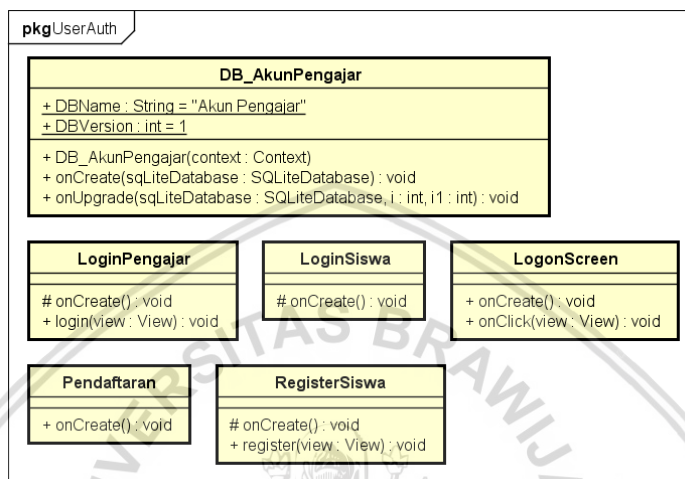
Paket UserAuth memiliki tiga klas yaitu klas LoginSiswa, LoginPengajar dan LogonScreen. Masing-masing dari klas tersebut digunakan pengguna untuk proses autentifikasi masuk ke dalam sistem. Perancangan klas diagram paket UserAuth dapat dilihat pada Gambar 5.25.



Gambar 5.25 Perancangan Class Diagram Paket UserAuth

5.1.3.14 Perancangan Class Diagram Paket UserAuth (Iterasi 1)

Paket UserAuth ini setelah terjadi perubahan kebutuhan terdapat enam buah kelas antara lain kelas LogonScreen, LoginPengajar, LoginSiswa, Pendaftaran dan RegisterSiswa sebagai *main class* dari masing-masing fitur, serta kelas DB_AkunPengajar digunakan untuk konfigurasi database akun pengajar. Perancangan klas diagram paket UserAuth (Iterasi 1) dapat dilihat pada Gambar 5.26.



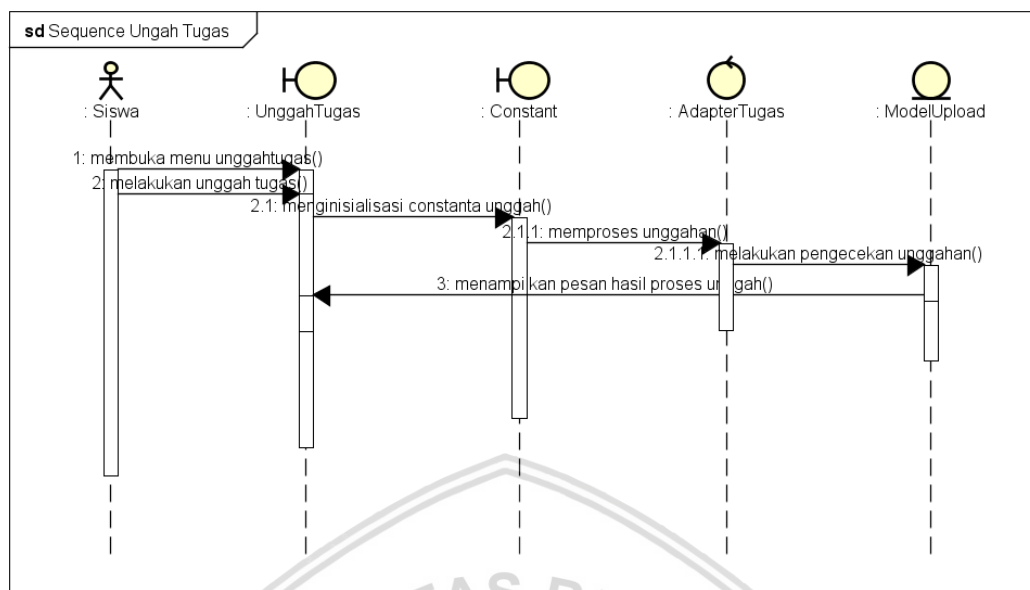
Gambar 5.26 Perancangan Class Diagram Paket UserAuth(Iterasi 1)

5.1.4 Perancangan Sequence Diagram

Sequence Diagram merupakan gambaran perilaku objek pada *use case diagram*. Diagram sekuen akan menggambarkan waktu hidup objek dan pesan antar objek, sehingga untuk menggambarkan diagram sekuen, objek-objek dalam *use case diagram* harus diketahui (Pratama, 2016). Diagram sekuen ini menggunakan metode *Model-View-Controller* (MVC) dalam penerapannya. Selain mengacu pada *use case diagram*, diagram sekuen juga melihat pada skenario diagram untuk menggambarkan pertukaran pesan dalam diagram sekuen.

5.1.4.1 Melakukan Unggah Berkas Tugas

Gambar 5.27 merupakan diagram *sequence* yang menggambarkan interaksi ketika siswa melakukan aktivitas unggah berkas tugas. Aktor pada diagram *sequence* ini adalah siswa yang akan melakukan aktivitas unggah tugas, saat aktor melakukan aktivitas unggah tugas, yang dilakukan adalah memilih tombol unggah tugas, kemudian sistem akan melakukan inisialisasi konstanta agar fitur unggah menggunakan *file manager* dapat dijalankan, saat proses unggah berjalan data akan dikirim menuju *AdapterTugas* dan *ModelUpload* agar berkas tugas dapat ditampilkan dalam bentuk *list* dan menyimpan berkas data tersebut dapat disimpan dalam *database* berdasarkan nama berkas dan url-nya. Kemudian setelah data selesai diproses, sistem akan menampilkan pesan berhasil atau tidaknya berkas tugas diunggah pada *database*.



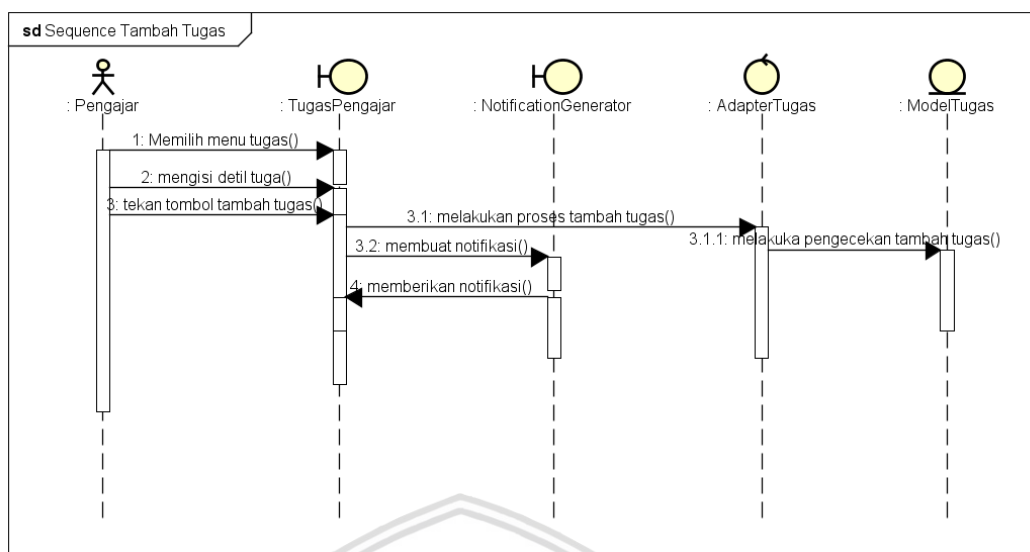
Gambar 5.27 Diagram Sequence Ungah Berkas Tugas

5.1.4.2 Melakukan Tambah Tugas

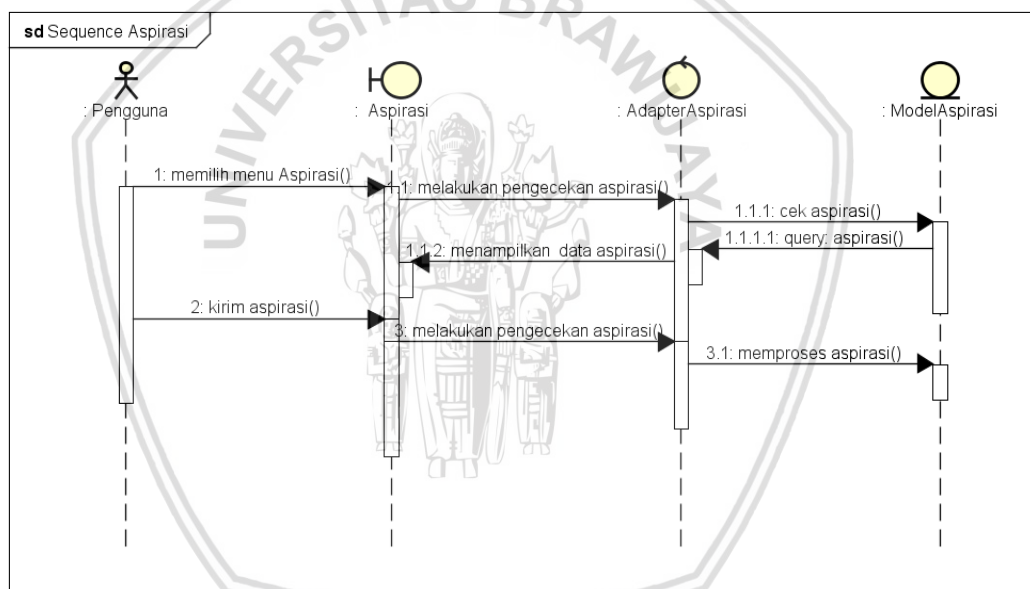
Gambar 5.28 merupakan diagram *sequence* yang menggambarkan interaksi ketika pengajar melakukan aktivitas tambah daftar tugas untuk siswa. Aktor pada diagram *sequence* ini merupakan pengajar. Tahap awal yang dilakukan aktor yaitu memilih menu tugas, kemudian aktor melakukan pengisian detil tugas yang akan diberikan kepada aktor siswa, saat aktor tekan tombol tambah tugas, sistem akan mengirimkan data yang dibuat aktor pengajar tadi agar diproses oleh AdapterTugas dan ModelTugas agar data yang dikirim tersebut dapat disimpan dan ditampilkan dalam bentuk *list*, kemudian saat aktor tekan tombol tamba tugas, sistem juga secara otomatis mengaktifkan *trigger* untuk membuat notifikasi pada NotificationGenerator, setelah itu notifikasi dapat muncul pada *device* aktor.

5.1.4.3 Melakukan Kirim dan Lihat Aspirasi

Gambar 5.29 merupakan diagram *sequence* yang menggambarkan interaksi ketika pengguna baik siswa maupun pengajar melakukan aktivitas pengiriman aspirasi dan melihat daftar aspirasi. Aktor pada diagram *sequence* ini merupakan pengguna siswa maupun pengajar. Saat akan melakukan pengiriman aspirasi, aktor memilih menu aspirasi dan memasukkan data aspirasi, kemudian data aspirasi akan dikirim ke AdapterAspirasi dan ModelAspirasi agar data dapat disimpan dan ditampilkan dalam bentuk *list*, kemudian data aspirasi dari aktor ditampilkan pada laman aspirasi.



Gambar 5.28 Diagram Sequence Tambah Daftar Tugas



Gambar 5.29 Diagram Sequence Kirim dan Lihat Aspirasi

5.1.5 Perancangan Algoritme

Perancangan algoritme ini berfungsi untuk menjelaskan langkah-langkah suatu proses pada beberapa fungsi tertentu. Pada bagian ini akan menjelaskan beberapa perancangan algoritme dari fungsi menampilkan pertanyaan latihan soal, menghitung skor latihan soal dan cetak data latihan soal.

5.1.5.1 Algoritme Menampilkan Pertanyaan Latihan Soal

Algoritme menampilkan pertanyaan latihan soal ini merupakan algoritme untuk melakukan proses menampilkan soal yang tersimpan pada *database* ke halaman aplikasi. Perancangan Algoritme Menampilkan Pertanyaan Latihan Soal dijelaskan pada Gambar 5.30.

5.1.5.2 Algoritme Perhitungan Skor Latihan Soal

Algoritme ini untuk menghitung skor pengguna saat melakukan latihan soal. Perancangan Algoritme ini dijelaskan pada Gambar 5.31.

5.1.5.3 Algoritme Cetak Data Skor Latihan Soal

Algoritme cetak data skor ini merupakan algoritme untuk melakukan proses mencetak skor latihan soal yang dijelaskan pada Gambar 5.32.

Algoritme perhitungan skor latihan soal**Deklarasi :**

- Button → pilih
- String → jawaban, kunci jawaban, skor

Deskripsi :

- Masukan: -
- Proses:
 1. Inisialisasi method perhitungan skor
 2. Inisialisasi tombol pilih jawaban
 3. Jika jawaban yang dipilih sama dengan kunci jawaban maka
 - a. Update variable skor dengan melakukan *increment* bertambah 1
 - b. Mencetak pesan jika jawaban yang dipilih benar
 4. Jika jawaban yang dipilih tidak sama dengan kunci jawaban maka Mencetak pesan jika jawaban yang dipilih salah Melakukan update skor sementara
- Keluaran: Menampilkan perolehan data skor latihan soal

Gambar 5.30 Algoritme Menampilkan Pertanyaan Latihan Soal

Algoritme menampilkan pertanyaan latihan soal

Deklarasi :

- Integer → nomorSoal, jumlahSoal

Deskripsi:

- Masukan: -
- Proses:
 1. Melakukan pengecekan apakah nomorSoal lebih besar dari jumlah soal
 2. Jika nomor soal lebih besar dari jumlah soal maka:
 - a. Menampilkan soal yang terdapat pada *database*
 3. Jika nomor soal lebih besar dari jumlah soal maka:
 - a. Menampilkan pesan bahwa latihan soal telah selesai
 - b. Menuju halaman skor latihan soal
 - c. Mencetak skor hasil latihan soal
- Keluaran: Pertanyaan akan muncul secara berurutan hingga nomor soal yang terdapat pada *database* sama dengan atau lebih dari jumlah soal yang terdapat pada *database*.

Gambar 5.31 Algoritme Perhitungan Skor Latihan Soal



Algoritme cetak data skor latihan soal**Deklarasi :**

- Integer → skor
- TextView → skorSekarang, skorTertinggi
- SharedPreferences → nilai

Deskripsi :

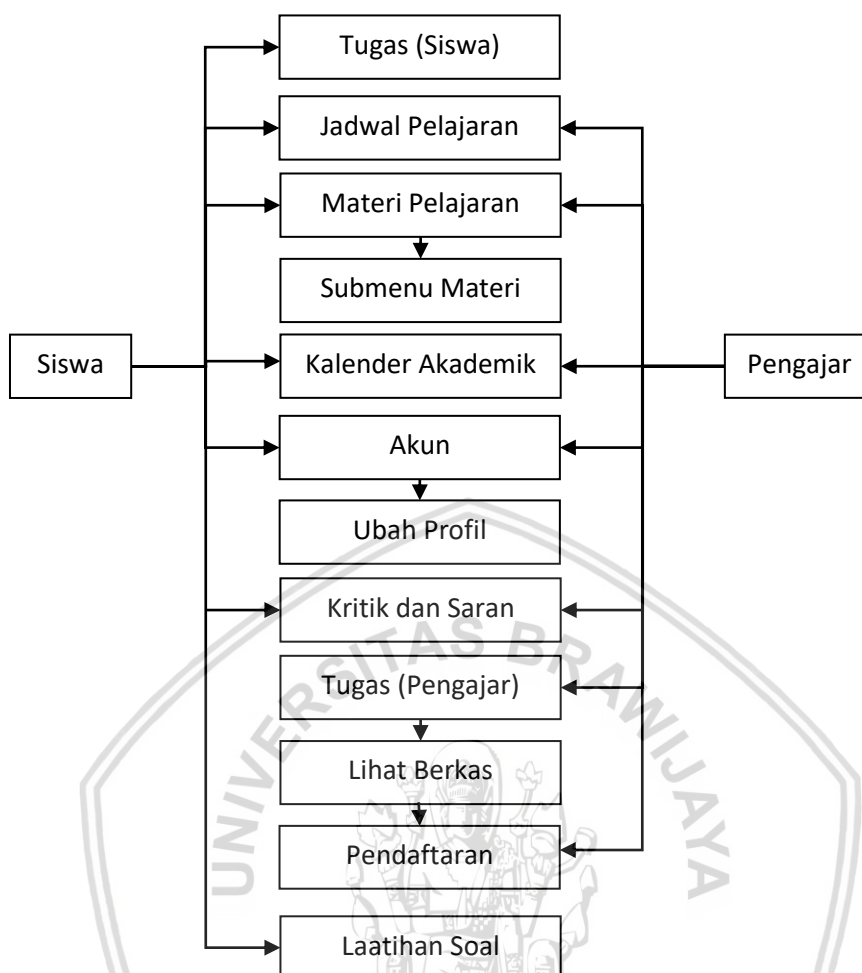
- Masukkan: -
- Proses:
 1. Inisialisasi method cetak data skor
 2. Inisialisasi variable skor, teks skorSekarang dan skorTertinggi
 3. Mencetak skor saat ini
 4. Inisialisasi nilai dengan menggunakan data sementara untuk skorTertinggi
 5. Jika skorTertinggi lebih tinggi dari skorSekarang maka:
 - a. Set teks skorTertinggi dengan dari nilai skorSekarang
 - b. Melakukan edit database sementara untuk menyimpan poin tertinggi
 6. Jika skorTertinggi tidak lebih tinggi dari skorSekarang maka:
 - a. Set teks skorSekarang dari nilai skor yang didapat
- Keluaran: nilai skor sekarang dan nilai skor tertinggi yang didapat

Gambar 5.32 Algoritme Cetak Data Skor Latihan Soal**5.1.6 Perancangan Antarmuka**

Perancangan antarmuka aplikasi E-learning SMANTIGDA akan dijelaskan pada bagian ini. Untuk perancangan antarmuka dibagi menjadi dua bagian yaitu perancangan *screenflow* Antarmuka dan perancangan detil antarmuka.

5.1.6.1 Perancangan Screenflow Antarmuka

Pada bagian ini akan dijelaskan bagaimana alur antarmuka dari aplikasi yang akan digunakan oleh pengguna. Serta dijelaskan arah tujuan untuk setiap halaman dari semua elemen (*button*) jika diberikan aksi.



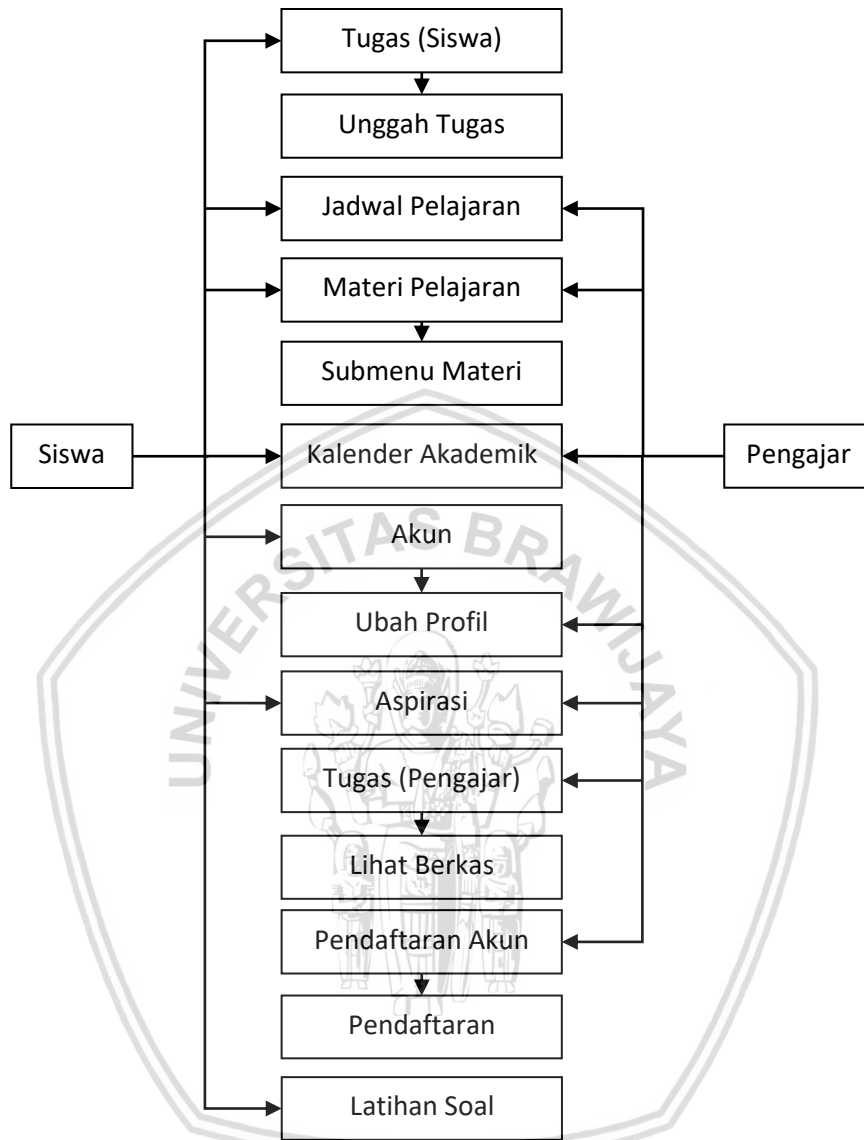
Gambar 5.33 Screenflow Aplikasi

Pada *screenflow* diatas diawali dari halaman *login screen*, kemudian masuk dengan masing-masing kategori, dari siswa akan dapat mengakses menu tugas (siswa), jadwal pelajaran, materi pelajaran, submenu materi, kalender, akademik, akun, ubah profil dan kritik dan Saran. Untuk pengajar dapat mengakses jadwal pelajaran, materi pelajaran, submenu materi, kalender akademik, akun, ubah profil, kritik & saran, tugas (pengajar), lihat berkas, pendaftaran akun, dan pendaftaran.

5.1.6.2 Perancangan Screenflow Antarmuka (Iterasi 1)

Adanya sedikit perubahan pada *screenflow* aplikasi ini, pergantian nama Kritik & Saran menjadi Aspirasi, juga terdapat penambahan halaman antarmuka unggah tugas untuk siswa dan pendaftarakn akun siswa untuku pengajar sehingga secara keseluruhan *screenflow* diawali dari halaman *login screen*, kemudian masuk dengan masing-masing kategori, dari siswa akan dapat mengakses menu tugas (siswa), unggah tugas, jadwal pelajaran, materi pelajaran, submenu materi, kalender akademik, akun, ubah profil dan aspirasi. Untuk pengajar dapat mengakses jadwal pelajaran, materi pelajaran, submenu materi, kalender akademik, akun, ubah profil, aspirasi, tugas (pengajar), lihat berkas,

pendaftaran akun, dan pendaftaran. Perancangan *Screenflow* Antarmuka Iterasi 1 ditunjukkan pada Gambar 5.34.

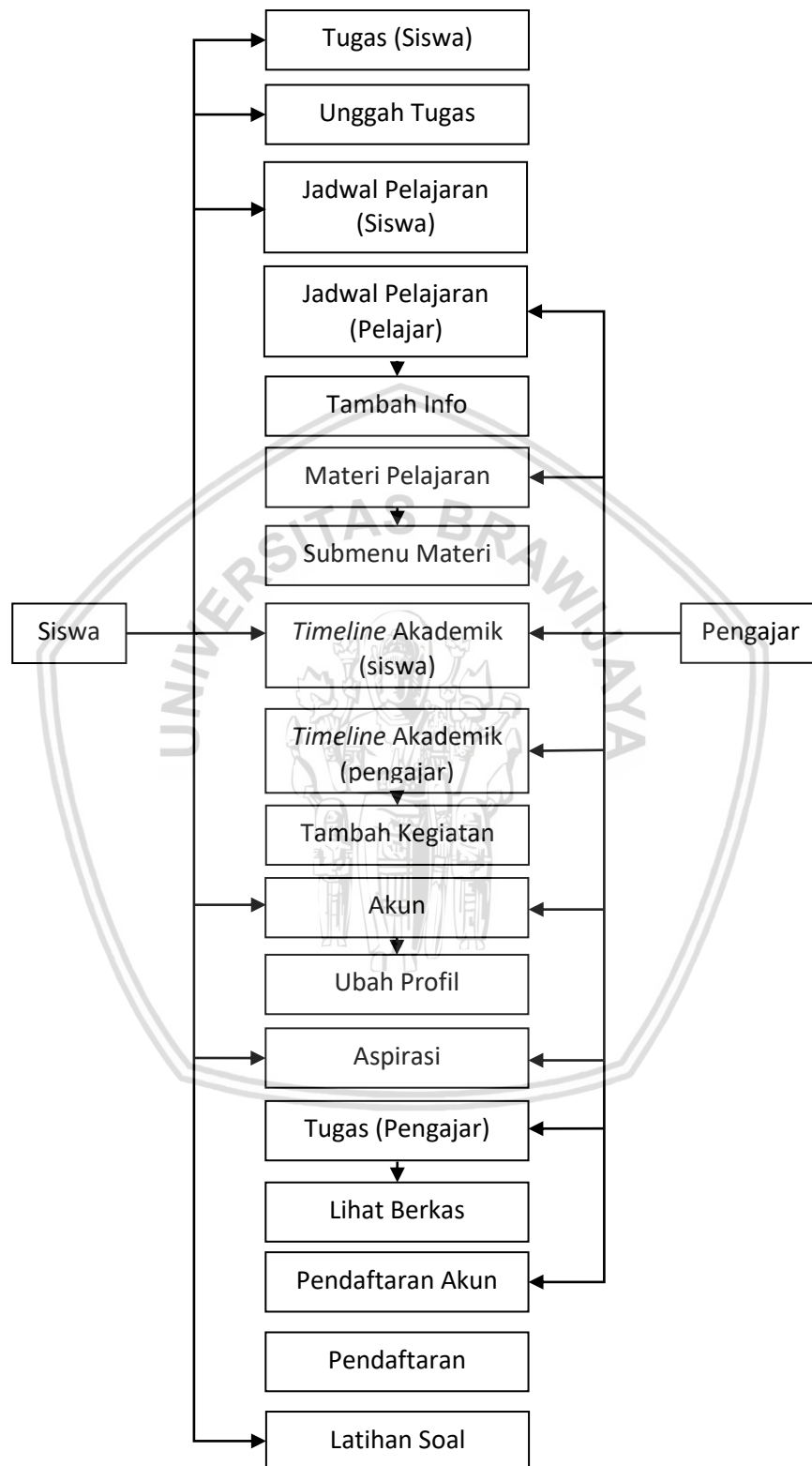


Gambar 5.34 Screenflow Aplikasi (Iterasi 1)

5.1.6.3 Perancangan *Screenflow* Antarmuka (Iterasi 2)

Terjadi perubahan lagi pada *screenflow* aplikasi ini, terdapat halaman antarmuka baru yaitu tambah kegiatan untuk pengajar serta pergantian nama kalender akademik menjadi *timenline* akademik sehingga secara keseluruhan *screenflow* diawali dari halaman *logon screen*, kemudian masuk dengan masing-masing kategori, dari siswa akan dapat mengakses menu tugas (siswa), unggah tugas, jadwal pelajaran, materi pelajaran, submenu materi, *timenline* akademik, akun, ubah profil dan aspirasi. Untuk pengajar dapat mengakses jadwal pelajaran, materi pelajaran, submenu materi, *timline* akademik, tambah kegiatan, akun, ubah profil, aspirasi, tugas (pengajar), lihat berkas, pendaftaran

akun, dan pendaftaran. Perancangan *Screenflow* Antarmuka Iterasi 1 ditunjukkan pada Gambar 5.35.



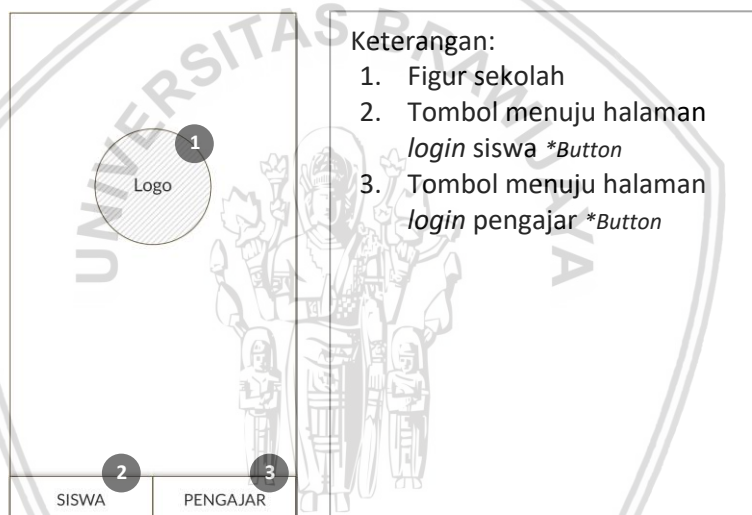
Gambar 5.35 *Screenflow* Aplikasi (Iterasi 2)

5.1.6.4 Perancangan Detil Antarmuka

Detil dari Perancangan antarmuka pada tiap halaman akan dijelaskan pada bagian ini. Terdapat beberapa halaman yang akan dijelaskan antara lain halaman *Logon Screen*, *Login*, Tugas (siswa), Unggah tugas, Jadwal Pelajaran (siswa), Jadwal Pelajaran (pengajar), Materi Pelajaran, Submenu Materi, Timeline Akademik, Tambah kegiatan, Akun, Ubah Profil, Asprasi, Tugas (pengajar), Lihat berkas, Pendaftaran Akun, Pendaftaran.

(a) Detil Perancangan Antarmuka *Logon Screen*

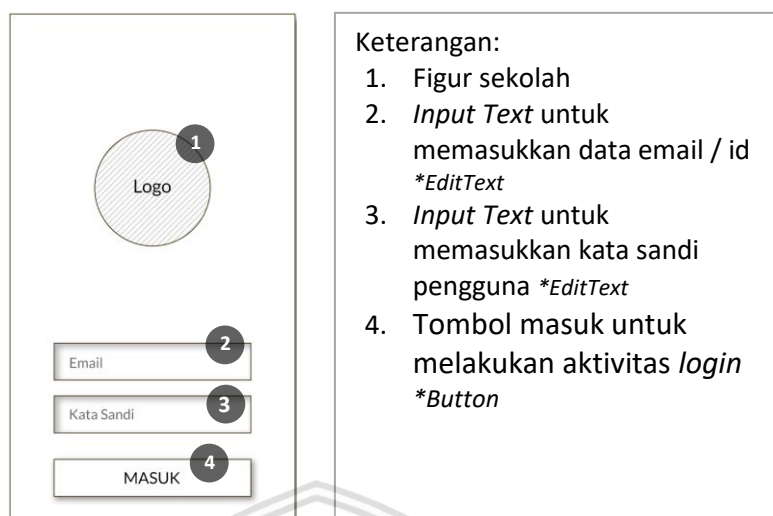
Pada laman ini terdapat tiga komponen penyusun yaitu gambar logo, dan dua tombol untuk menuju ke laman *login*. Pada laman ini pengguna diminta untuk memilih kategori *login*, masuk sebagai siswa atau masuk sebagai pengajar. Gambar 5.36 menunjukkan rancangan antarmuka halaman *Logon Screen*.



Gambar 5.36 Rancangan Antarmuka *Logon Screen*

(b) Detil Perancangan Antarmuka *Login*

Halaman *login* siswa dan *login* pengajar memiliki komponen penyusun yang sama, yaitu form untuk memasukkan data berupa *email/username* dan kata sandi, kemudian terdapat tombol masuk untuk menuju ke halaman menu utama. Gambar 5.37 menunjukkan rancangan antarmuka halaman *login*.



Gambar 5.37 Rancangan Antarmuka Login Screen

(c) Detil Perancangan Antarmuka Halaman Utama Siswa

Pada halaman menu utama siswa terdapat beberapa menu utama antara lain menu jadwal pelajaran, menu materi pelajaran, menu tugas, menu kalender akademik, menu aspirasi dan menu akun. Gambar 5.38 menunjukkan rancangan antarmuka menu utama siswa.



Gambar 5.38 Rancangan Antarmuka Halaman Utama Siswa

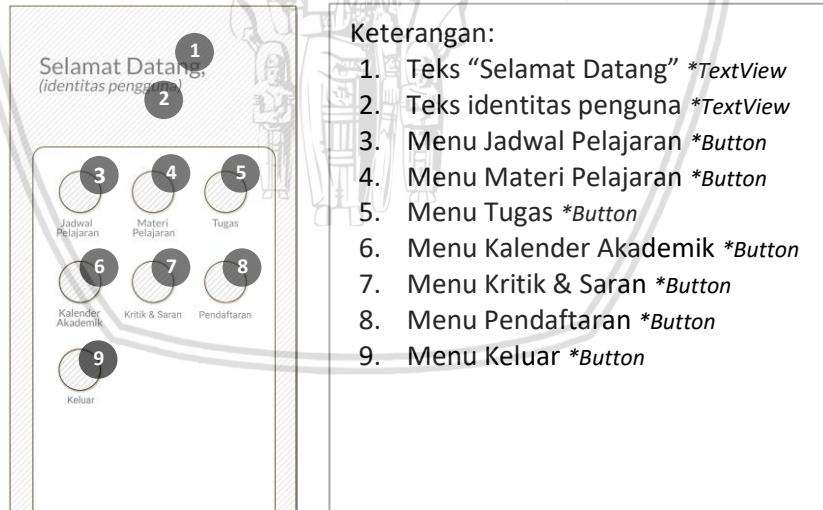
Terjadi perubahan nama menu kalender akademik menjadi timeline akademik, kemudian kritik & saran menjadi aspirasi. Gambar 5.39 menunjukkan rancangan antarmuka halaman utama pengajar iterasi 1.



Gambar 5.39 Rancangan Antarmuka Halaman Utama Siswa (Iterasi 1)

(d) Detil Perancangan Antarmuka Halaman Utama Pengajar

Pada halaman menu utama siswa terdapat beberapa menu utama antara lain menu jadwal pelajaran, menu materi pelajaran, menu tugas, menu kalender akademik, menu aspirasi, menu akun dan menu pendaftaran. Gambar 5.40 menunjukkan rancangan antarmuka menu utama siswa



Gambar 5.40 Rancangan Antarmuka Halaman Utama Pengajar

Terjadi perubahan nama menu kalender akademik menjadi timeline akademik, kemudian kritik & saran menjadi aspirasi. Gamabr 5.41 menunjukkan rancangan antarmuka halaman utama pengajar iterasi 1.



Gambar 5.41 Rancangan Antarmuka Halaman Utama Pengajar (Iterasi 1)

(e) Detil Perancangan Antarmuka Tugas (Siswa)

Rancangan antarmuka halaman tugas siswa menampilkan daftar tugas yang telah diberikan oleh pengajar, detail dari daftar tugas terdapat materi pelajaran, tema tugas pelajaran, deskripsi tugas, index tugas, kelas, dan informasi *deadline* pengumpulan tugas. Kemudian pada bagian bawah kanan terdapat tombol untuk menuju halaman pengumpulan tugas. Gambar 5.42 menunjukkan rancangan antarmuka tampilan latihan soal dan tugas

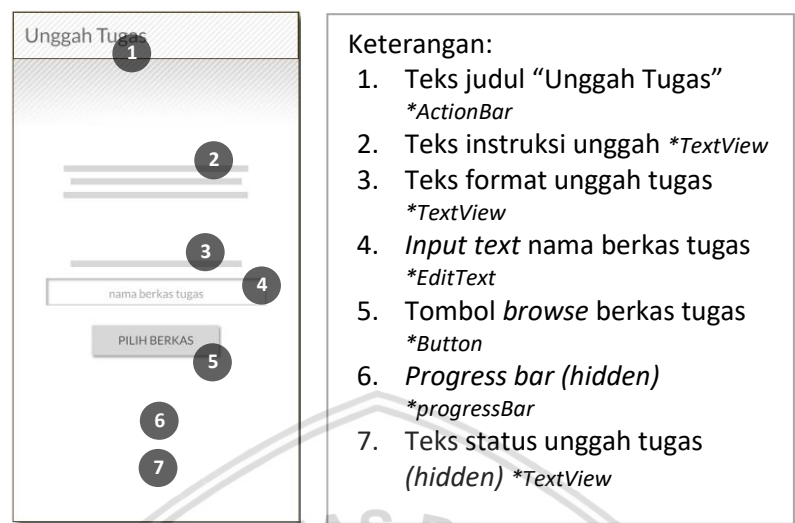


Gambar 5.42 Rancangan Antarmuka Tugas (Siswa)

(f) Detil Perancangan Antarmuka Unggah Tugas

Antarmuka unggah tugas ini berada pada aplikasi sisi siswa, pada halaman ini terdapat komponen EditText untuk memberikan nama berkas tugas sebelum diunggah. Kemudian juga terdapat tombol untuk memilih berkas

tugas yang ada pada memori penyimpanan *smartphone*. Gambar 5.43 menunjukkan rancangan antarmuka unggah tugas.



Gambar 5.43 Rancangan Antarmuka Unggah Tugas

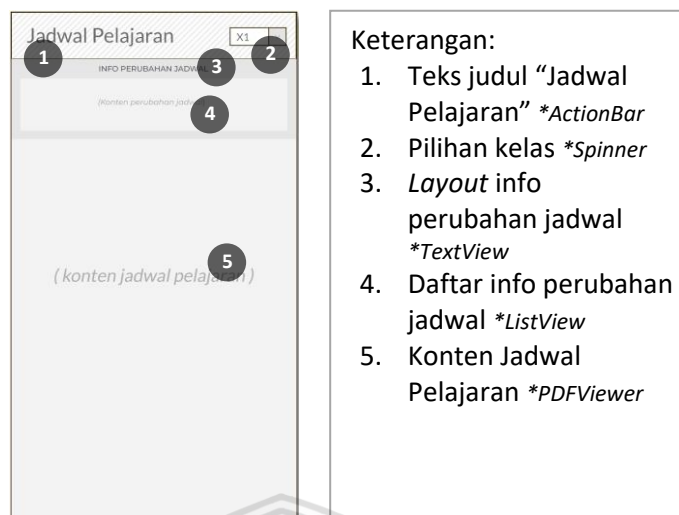
(g) Detil Perancangan Antarmuka Jadwal Pelajaran (Siswa)

Halaman jadwal pelajaran terdapat dua komponen yaitu spinner untuk pilihan kelas dan layout konten jadwal pelajaran itu sendiri, spinner untuk pilihan kelas berada pada toolbar dari activity jadwal pelajaran siswa dan pengajar. Gambar 5.44 menunjukkan rancangan antarmuka halaman jadwal pelajaran siswa.



Gambar 5.44 Rancangan Antarmuka Jadwal Pelajaran Siswa

Terjadi perubahan pada halaman perancangan antarmuka tugas siswa karena terdapat penambahan fungsi baru yaitu info perubahan jadwal pada bagian atas layar setelah *action bar*. Gambar 5.45 menunjukkan rancangan antarmuka halaman jadwal pelajaran siswa.



Gambar 5.45 Rancangan Antarmuka Jadwal Pelajaran Siswa (Iterasi 1)

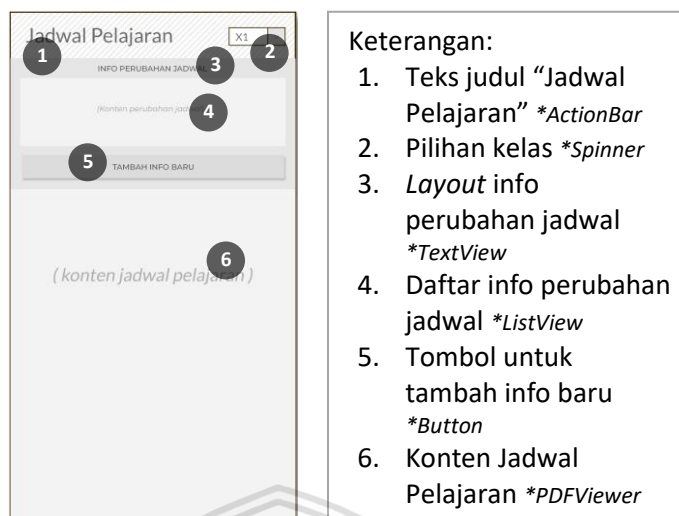
(h) Detil Perancangan Antarmuka Jadwal Pelajaran (Pengajar)

Halaman jadwal pelajaran awalnya sama seperti halaman jadwal pelajaran pada halaman siswa, terdapat dua komponen yaitu spinner untuk pilihan kelas dan layout konten jadwal pelajaran itu sendiri, spinner untuk pilihan kelas berada pada toolbar dari activity jadwal pelajaran pengajar. Gambar 5.46 menunjukkan rancangan antarmuka halaman jadwal pelajaran siswa.



Gambar 5.46 Rancangan Antarmuka Jadwal Pelajaran Pengajar

Namun terjadi perubahan pada halaman perancangan antarmuka tugas siswa karena terdapat penambahan fungsi baru yaitu info perubahan jadwal pada bagian atas halaman setelah *action bar*, yang membedakan dengan halaman jadwal pelajaran pada siswa adalah adanya tombol untuk menuju halaman tambah info perubahan jadwal baru. Gambar 5.47 menunjukkan rancangan antarmuka halaman jadwal pelajaran pengajar.



Gambar 5.47 Rancangan Antarmuka Jadwal Pelajaran Pengajar (Iterasi 1)

(i) Detil Perancangan Antarmuka Tambah Info

Halaman tambah info terdapat pada sisi pengajar, terdapat *form* untuk memasukkan data text info perubahan jadwal, kemudian dibawah form tersebut terdapat tiga spinner tanggal, bulan dan tahun untuk menunjukkan tanggal perubahan jadwal tersebut dibuat, kemudian juga terdapat tombol simpan untuk melakukan pembaharuan info. Gambar 5.48 menunjukkan rancangan antarmuka tambah info.

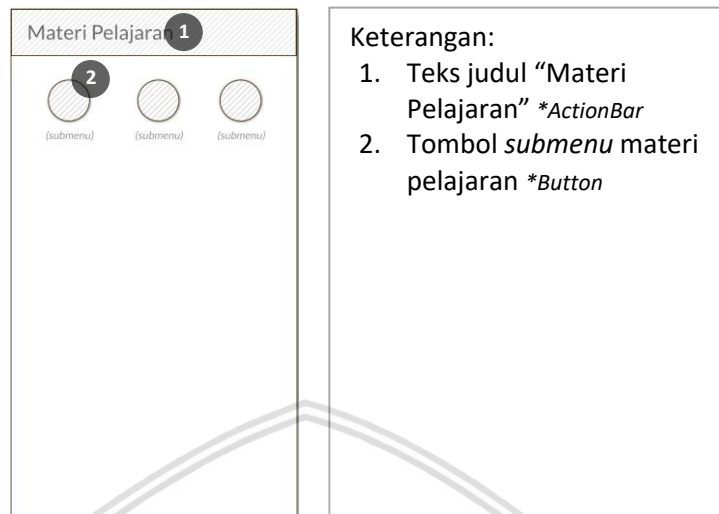


Gambar 5.48 Rancangan Antarmuka Tambah Info

(j) Detil Perancangan Antarmuka Submenu Materi

Halaman Sub-Materi Pelajaran akan ditampilkan aplikasi saat pengguna memilih menu Materi Pelajaran, sebelum masuk ke halaman utama materi pelajaran, aplikasi akan menampilkan halaman sub-materi pelajaran untuk memilih kelas dan keminatan untuk menuju halaman materi pelajaran yang

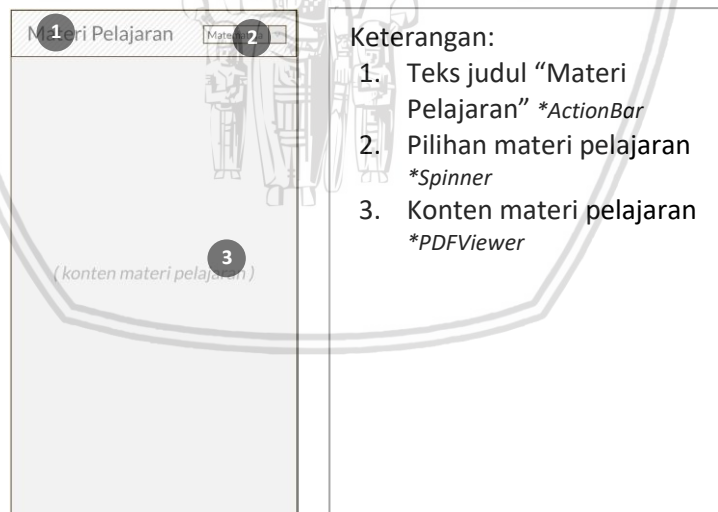
lebih spesifik. Gambar 5.49 menunjukkan rancangan antarmuka halaman submenu materi pelajaran.



Gambar 5.49 Rancangan Antarmuka Submenu Materi

(k) Detil Perancangan Antarmuka Materi Pelajaran

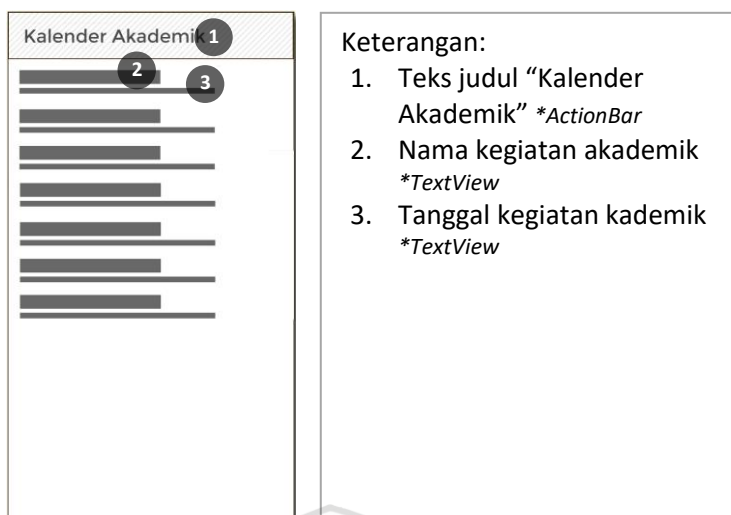
Halaman Materi Pelajaran memiliki dua komponen penyusun, yaitu *spinner* pada *actionbar* dan layout materi pelajaran itu sendiri. *Spinner* digunakan untuk memilih materi pelajaran yang tersedia. Gambar 5.50 menunjukkan rancangan antarmuka halaman materi pelajaran.



Gambar 5.50 Rancangan Antarmuka Materi Pelajaran

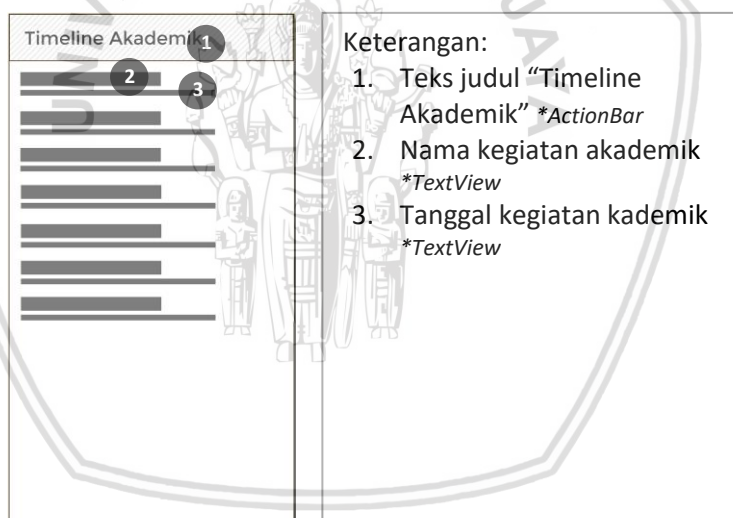
(l) Detil Perancangan Antarmuka Timeline Akademik (Siswa)

Halaman kalender akademik menampilkan *timeline* kalender akademik per-tahun-nya yang dipisahkan berdasarkan bulan. Gambar 5.51 menunjukkan rancangan antarmuka kalender akademik.



Gambar 5.51 Rancangan Antarmuka Timeline Akademik Siswa

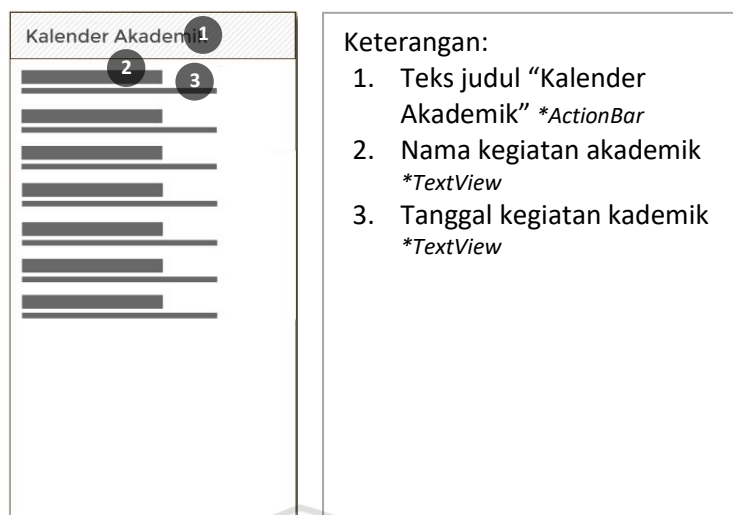
Halaman kalender akademik terjadi perubahan judul pada *actionbar*-nya, perubahan judul tersebut dikarenakan isi dari konten menu tersebut lebih mengarah ke bentuk *timeline* daripada ke bentuk kalender. Gambar 5.52 menunjukkan rancangan antarmuka timeline akademik Siswa (iterasi 1).



Gambar 5.52 Rancangan Antarmuka Timeline Akademik Siswa (Iterasi 1)

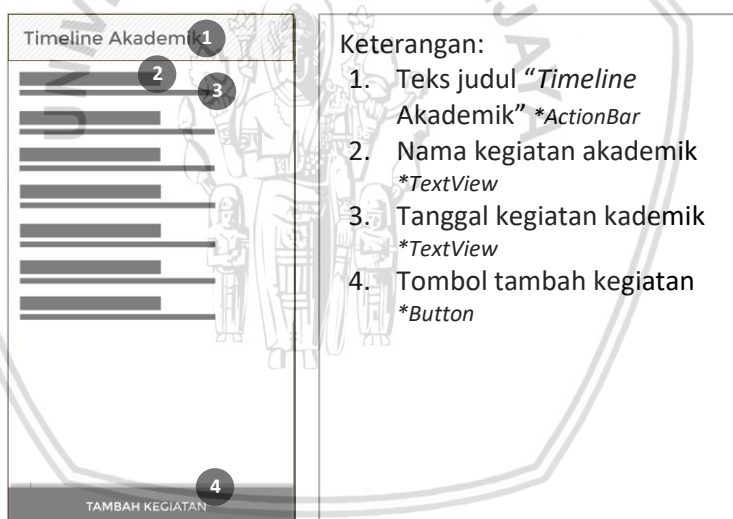
(m) Detil Perancangan Antarmuka Timeline Akademik (Pengajar)

Tidak berbeda dengan *timeline* akademik siswa, kalender akademik menampilkan *timeline* kalender akademik per-tahun-nya yang dipisahkan berdasarkan bulan. Gambar 5.53 menunjukkan rancangan antarmuka kalender akademik.



Gambar 5.53 Rancangan Antarmuka Timeline Akademik Pengajar

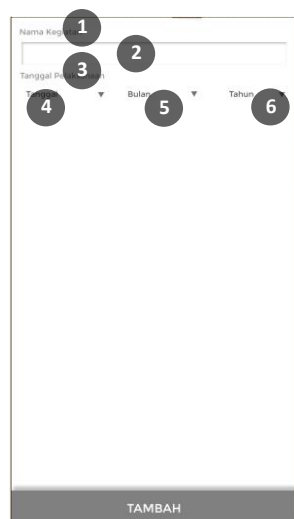
Setelah terjadi perubahan, ada tambahan satu komponen pada halaman perancangan timeline akademik pengajar, yaitu terdapat tombol untuk menuju ke halaman tambah kegiatan. Gambar 5.54 menunjukkan rancangan antarmuka timeline akademik pengajar iterasi 1.



Gambar 5.54 Rancangan Antarmuka Timeline Akademik Pengajar(Iterasi 1)

(n) Detil Perancangan Antarmuka Tambah Kegiatan

Halaman tambah kegiatan ini terdapat beberapa komponen yaitu *form* untuk melakukan pengisian data *text* judul kegiatan, *spinner* untuk masukan data tanggal, bulan dan tahun, kemudian pada bagian bawah halaman terdapat tombol tambah untuk menambah kegiatan baru pada timeline akademik. Gambar 5.55 menunjukkan rancangan antarmuka tambah kegiatan.



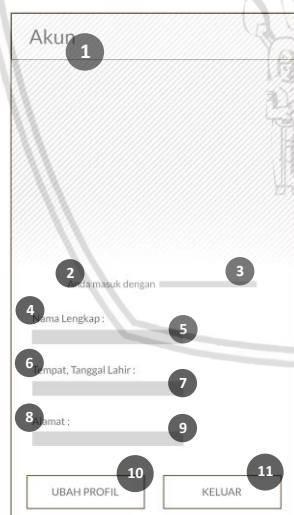
Keterangan:

1. Teks judul "Tambah Kegiatan Akademik" *ActionBar
2. Teks "Nama Kegiatan" *TextView
3. Input text nama kegiatan *EditText
4. Teks "Tanggal Pelaksanaan" *TextView
5. Pilihan tanggal kegiatan *Spinner
6. Pilihan bulan kegiatan *Spinner
7. Pilihan tahun kegiatan *Spinner
8. Tombol tambah *Button

Gambar 5.55 Rancangan Antarmuka Tambah Kegiatan

(o) Detil Perancangan Antarmuka Akun

Halaman akun menampilkan data akun yang sedang masuk pada sistem saat aplikasi dijalankan. Terdapat data *email*, nama lengkap, tempat tanggal lahir dan alamat pengguna. Pada bagian bawah juga terdapat dua tombol, tombol ubah akun untuk memperbaharui data pengguna, dan tombol keluar untuk mengakhiri sesi pengguna pada aplikasi. Gambar 5.56 menunjukkan perancangan antarmuka akun.



Keterangan:

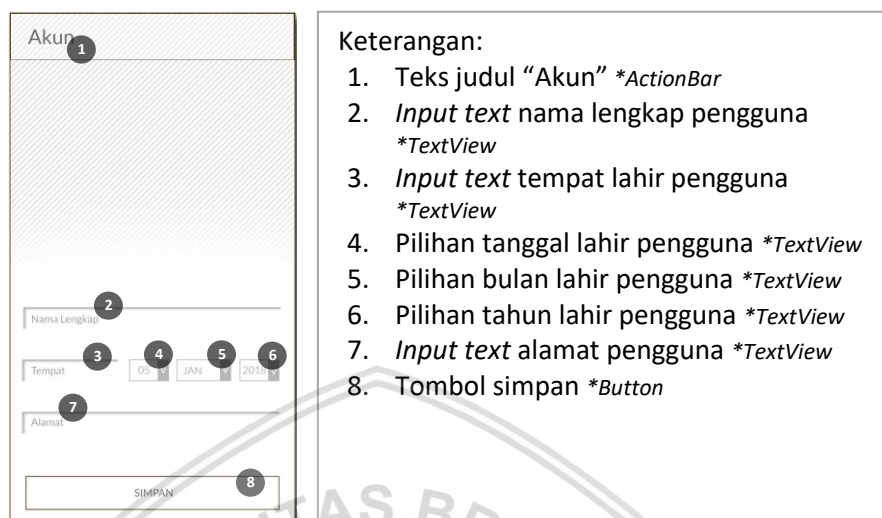
1. Teks judul "Akun" *ActionBar
2. Teks "Anda masuk dengan" *TextView
3. Teks *email* pengguna *TextView
4. Teks "Nama Lengkap:" *TextView
5. Teks nama lengkap pengguna *TextView
6. Teks "Tempat Tanggal Lahir:" *TextView
7. Teks tempat tanggal lahir pengguna *TextView
8. Teks "Nama Lengkap:" *TextView
9. Teks alamat pengguna *TextView
10. Tombol ubah profil *Button
11. Tombol keluar *Button

Gambar 5.56 Rancangan Antarmuka Akun

(p) Detil Perancangan Antarmuka Ubah Profil

Halaman ubah profil digunakan pengguna untuk memperbaharui data akunnya dalam aplikasi, terdapat tiga macam komponen yaitu *EditText* untuk memperbaharui nama lengkap pengguna, tanggal lahir pengguna, dan alamat pengguna, kemudian terdapat tiga buah spinner untuk memperbaharui data tanggal, bulan, dan tahun lahir pengguna. Dan pada

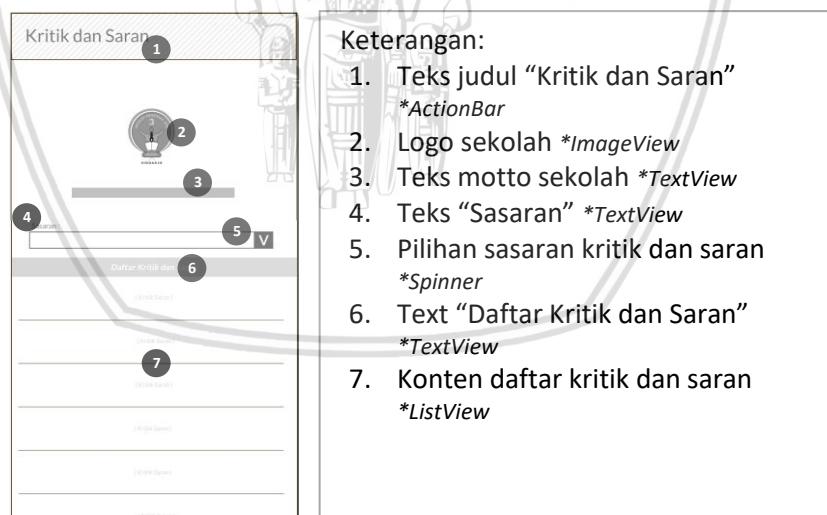
bagian bawah halaman terdapat satu buah tombol simpan untuk menyimpan perubahan data pengguna. Gambar 5.57 menunjukkan rancangan antarmuka ubah profil.



Gambar 5.57 Rancangan Antarmuka Ubah Profil

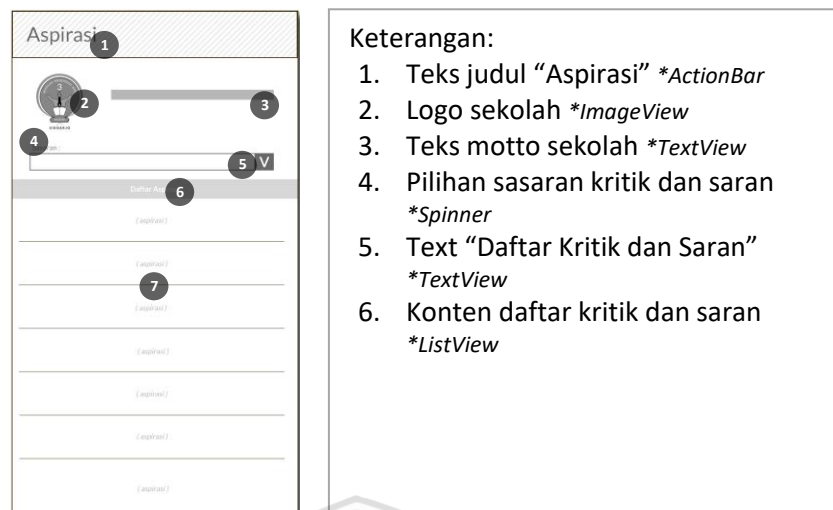
(q) Detil Perancangan Antarmuka Aspirasi

Halaman Aspirasi menampilkan logo sekolah, teks ajakan untuk menyuarakan aspirasi, sasaran aspirasi dan daftar aspirasi yang telah masuk pada database. Gambar 5.58 menunjukkan rancangan antarmuka aspirasi.



Gambar 5.58 Rancangan Antarmuka Aspirasi

Adanya perubahan perancangan antarmuka pada antarmuka Kritik dan Saran, perubahan terjadi pada judul aktivitas menjadi Aspirasi, serta komposisi *layout* pada halaman juga terjadi perubahan, terjadi perluasan *layout* daftar aspirasi. Berikut rancangan antarmuka Aspirasi ditunjukkan pada gambar 5.59.



Keterangan:

1. Teks judul "Aspirasi" *ActionBar
2. Logo sekolah *ImageView
3. Teks motto sekolah *TextView
4. Pilihan sasaran kritik dan saran *Spinner
5. Text "Daftar Kritik dan Saran" *TextView
6. Konten daftar kritik dan saran *ListView

Gambar 5.59 Rancangan Antarmuka Aspirasi (Iterasi 1)

(r) Detil Perancangan Antarmuka Tugas (Pengajar)

Halaman tugas pengajar terdapat empat jenis komponen penyusun, *EditText*, *Spinner*, *Button*, dan *ListView*. Untuk pilihan mata pelajaran, kelas, jam dan tanggal pengumpulan menggunakan *spinner* untuk menampilkan pilihannya. Kemudian untuk tema *index* dan deskripsi tugas menggunakan *EditText* untuk mendapatkan data dari penggunaanya. Terdapat dua tombol yaitu tombol tambah untuk mengirimkan tugas ke siswa dan lihat berkas untuk melihat daftar tugas yang telah dikirimkan oleh siswa ke *database*. Pada bagian bawah halaman terdapat *ListView* yang digunakan untuk menampilkan daftar tugas yang diberikan oleh seluruh pengajar. Gambar 5.60 menunjukkan rancangan antarmuka halaman tugas Pengajar.

Keterangan:

1. Teks judul "Tugas" *ActionBar
2. Teks "Mata Pelajaran" *TextView
3. Pilihan mata pelajaran *Spinner
4. Teks "Tema Tugas" *TextView
5. Input text tema tugas *EditText
6. Teks "Index Tugas" *TextView
7. Input text indeks tugas *EditText
8. Teks "Kelas" *TextView
9. Pilihan sasaran tugas kelas *Spinner
10. Teks "Jam" *TextView
11. Pilihan jam pengumpulan tugas *Spinner
12. Pilihan meridiem pengumpulan tugas *Spinner
13. Teks "Tanggal" *TextView
14. Pilihan tanggal pengumpulan tugas *Spinner
15. Pilihan bulan pengumpulan tugas *Spinner
16. Pilihan tahun pengumpulan tugas *Spinner
17. Teks "Deskripsi tugas" *TextView
18. Input text deskripsi tugas *EditText
19. Tombol tambah tugas *Button
20. Tombol lihat berkas tugas *Button
21. Konten daftar tugas *ListView

Gambar 5.60 Rancangan Antarmuka Tugas (Pengajar)

(s) Detil Perancangan Antarmuka Lihat Berkas

Halaman lihat berkas menampilkan daftar berkas yang diunggah oleh siswa untuk pengajar, halaman ini hanya dapat diakses oleh pengajar. Gambar 5.61 menunjukkan rancangan antarmuka halaman lihat berkas.

Keterangan:

1. Teks judul "Tugas" *ActionBar
2. Konten daftar berka tugas *ListView

Gambar 5.61 Rancangan Antarmuka Lihat Berkas

(t) Detil Perancangan Antarmuka Pendaftaran Akun

Halaman pendaftaran akun berada pada aplikasi sisi pengajar, pada halaman ini pengajar diminta untuk memilih kategori akun pendaftaran, yaitu akun siswa atau akun pengajar. Gambar 5.62 menjelaskan tentang perancangan antarmuka pendaftaran akun.



Gambar 5.62 Rancangan Antarmuka Pendaftaran Akun

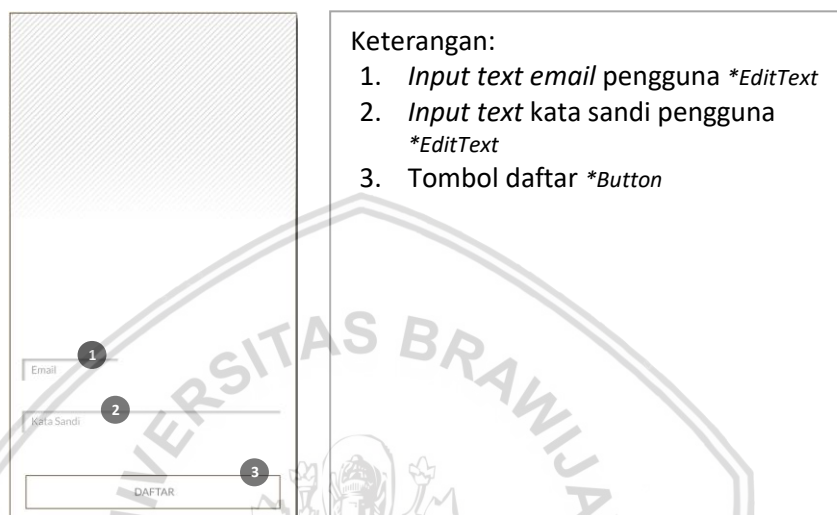
Terjadi perubahan pada salah satu fungsi pendaftaran, yaitu dihapuskannya pendaftaran akun pengajar karena akun pengajar bersifat statis dan tetap, serta menghindari terjadinya tercampurnya akun siswa dan pengajar dalam satu *database*, maka pendaftaran akun hanya memiliki satu opsi pilihan yaitu pendaftaran. Gambar 5.63 menjelaskan tentang rancangan antarmuka pendaftaran akun iterasi 1.



Gambar 5.63 Rancangan Antarmuka Pendaftaran Akun (Iterasi 1)

(u) Detil Perancangan Antarmuka Pendaftaran

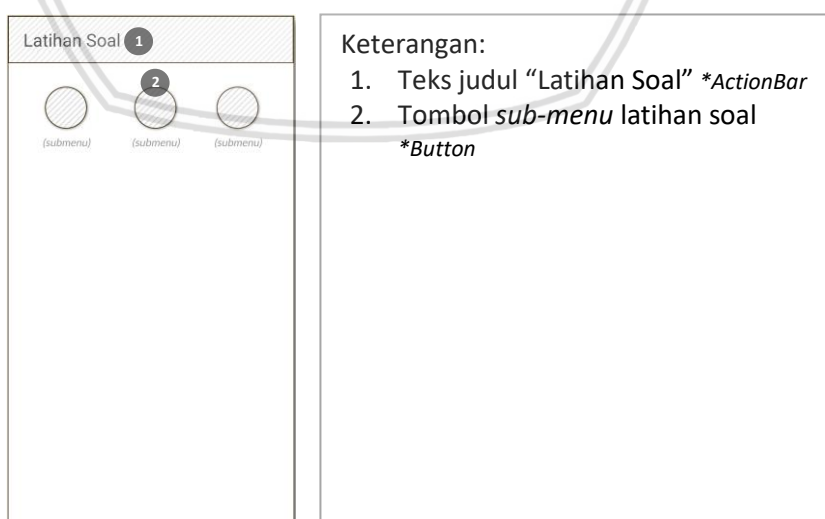
Halaman pendaftaran akun ini hanya terdapat pada sisi pengajar saja, baik pendaftaran akun untuk siswa dan pengajar. Pada halaman ini terdapat dua buah komponen, yaitu *EditText* dan *Button*. *EditText* digunakan untuk mengambil data masukan dari pengguna berupa email dan kata sandi, kemudian *Button* untuk melakukan pengiriman data ke *database*. Gambar 5.64 menjelaskan tentang rancangan antarmuka pendaftaran.



Gambar 5.64 Rancangan Antarmuka Pendaftaran

(v) Detil Perancangan Antarmuka Menu Latihan Soal

Pada halaman menu latihan soal terdapat komponen tombol untuk mengarahkan ke latihan soal yang lebih spesifik, pada menu latihan soal pengguna diminta untuk memilih kelas dan mata pelajaran. Gambar 5.65 menjelaskan tentang rancangan antarmuka menu latihan soal.



Gambar 5.65 Rancangan Antarmuka Menu Latihan Soal

(w) Detil Perancangan Antarmuka Latihan Soal

Pada halaman latihan soal terdapat teks soal pada bagian atas, kemudian dibawah teks soal terdapat empat buah tombol dengan teks opsi jawaban yang ada. Pada bagian bawah halaman terdapat perolehan poin sementara pengguna. Gambar 5.66 menjelaskan rancangan antarmuka latihan soal.



Gambar 5.66 Rancangan Antarmuka Latihan Soal

(x) Detil Perancangan Antarmuka Hasil Latihan Soal

Pada halaman hasil latihan soal menampilkan poin yang diraih oleh pengguna, baik perolehan poin pada saat itu atau poin tertinggi yang didapat. Selain itu juga terdapat nama pengguna dan dua buah tombol pada bagian bawah halaman yang berfungsi untuk memulai latihan soal kembali dan tombol kembali ke menu aplikasi. Gambar 5.67 menjelaskan rancangan antarmuka hasil latihan soal.



Gambar 5.67 Rancangan Antarmuka Hasil Latihan Soal

5.2 Implementasi

Pada sub-bab implementasi menjelaskan tentang implementasi yang dilakukan dalam pembuatan aplikasi. Pada bagian ini akan dijelaskan terlebih dahulu tentang spesifikasi sistem dan batasan implementasi sistem yang digunakan dalam tahap implementasi. Selanjutnya juga akan membahas implementasi basis data, implementasi *class*, implementasi kode program dan implementasi antarmuka.

5.2.1 Spesifikasi Sistem

Hasil dari analisis kebutuhan, perancangan dan hasil evaluasi akan diimplementasikan pada sebuah perangkat keras dan membutuhkan perangkat lunak. Perangkat keras dan perangkat lunak tersebut dispesifikasikan agar dapat mengetahui standar yang diperlukan selama melakukan pembangunan aplikasi.

5.2.1.1 Spesifikasi Perangkat Keras

Dalam melakukan implementasi aplikasi E-Learning SMANTIGDA dibutuhkan perangkat keras berupa *laptop* dan *smartphone*. *Laptop* digunakan dalam melakukan pembangunan aplikasi, sementara untuk *smartphone* digunakan untuk melakukan pengujian terhadap aplikasi yang sedang dibangun. Spesifikasi perangkat keras *laptop* yang digunakan dalam melakukan implementasi dapat dilihat pada Tabel 5.8.

Tabel 5.8 Spesifikasi Perangkat Keras

Keterangan	Spesifikasi
<i>System Model</i>	Acer Aspire V5 471-G
<i>Processor</i>	Intel(R) Core(TM) i5-3371U CPU @ 1,70 GHz
<i>Memory</i>	6 GB RAM, 500GB HDD
<i>Display</i>	Nvidia GEFORCE GT 620M

5.2.1.2 Spesifikasi Perangkat Lunak

Sedangkan untuk spesifikasi perangkat lunak *laptop* dapat dilihat pada Tabel 5.9.

Tabel 5.9 Spesifikasi Perangkat Lunak

Acer Aspire V5 471-G	
<i>Operating System</i>	Microsoft Windows 10 Pro 64bit
<i>Programming Language</i>	Java
<i>Programming Tool</i>	Android Studio 3.1
<i>Android SDK Tools</i>	26.0.2
<i>Gradle version</i>	4.1

Database Management System	Firestore Realtime Database SDK 11.6.2 SQLite
----------------------------	--

5.2.1.3 Spesifikasi Perangkat Bergerak

Sedangkan untuk spesifikasi perangkat lunak *smartphone* dapat dilihat pada tabel 5.10.

Tabel 5.10 Spesifikasi Perangkat Bergerak

Keterangan	Spesifikasi
<i>System Model</i>	Sony Xperia XA Ultra – F3216
<i>Processor</i>	Mediatek MT6755 Helio P10, Octa-core (4x2.0 GHz Cortex-A53 & 4x1.0 GHz Cortex-A53), Mali-T860MP2
<i>Memory</i>	16 GB, 3 GB RAM
<i>Display</i>	1080 x 1920 pixels, 6.0" 16:9 ratio (~367 ppi density)

5.2.2 Batasan Implementasi

Dalam melakukan implementasi aplikasi terdapat beberapa batasan implementasi, yaitu sebagai berikut:

1. Aplikasi dijalankan pada *smartphone* dengan sistem operasi Android dengan API 19 keatas.
2. Aplikasi dirancang menggunakan Android Studio dengan bahasa pemrograman java.
3. Aplikasi dijalankan pada *smartphone* dengan koneksi internet.
4. Aplikasi menggunakan Firebase dan SQLite sebagai manajemen basis datanya.

5.2.3 Implementasi Basis Data

Manajemen basis data aplikasi E-Learning ini menggunakan *Firestore Realtime Database*. Seperti yang dijelaskan sebelumnya, *Firestore Realtime Database* merupakan manajemen basis data yang menggunakan NoSQL sehingga penjelasan implementasi *database* akan digambarkan dengan operasi *tree*.

5.2.3.1 Basis Data Akun Siswa

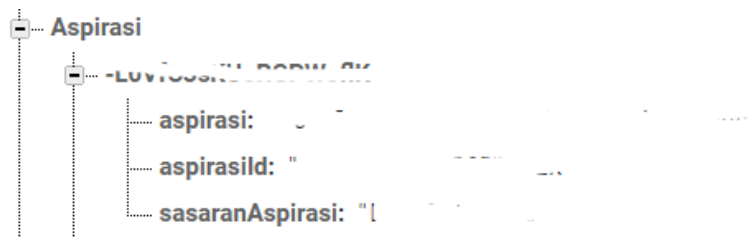
Implementasi basis data akun siswa terdapat beberapa atribut yaitu *alamatSiswa*, *namaSiswa*, *tanggalLahir*, *tempatLahir*. Basis data Akun Siswa dapat dilihat pada Gambar 5.68.



Gambar 5.68 Implementasi Basis Data Akun Siswa

5.2.3.2 Basis Data Aspirasi

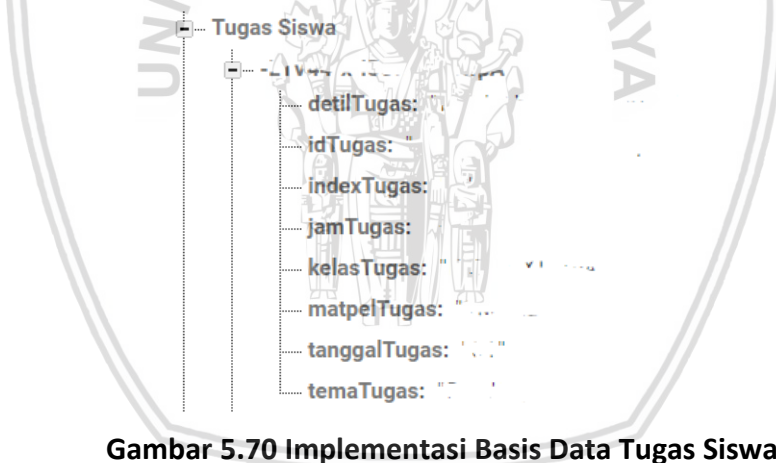
Implementasi basis data aspirasi terdapat beberapa atribut penyusunnya yaitu *aspirasi* untuk menyimpan aspirasi dari siswa dan pengajar, *aspirasiId* untuk menyimpan id aspirasi dan *sasaranAspirasi* untuk menyimpan tujuan atau sasaran aspirasi yang diberikan. Basis data Aspirasi Siswa dapat dilihat pada Gambar 5.69.



Gambar 5.69 Implementasi Basis Data Aspirasi

5.2.3.3 Basis Data Tugas Siswa

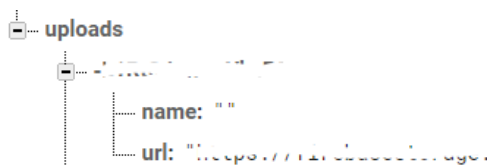
Implementasi basis data Tugas Siswa terdapat beberapa atribut penyusun yaitu *bulanTugas*, *detilTugas*, *idTugas*, *indexTugas*, *jamTugas*, *kelasTugas*, *matpelTugas*, *tanggalTugas*, dan *temaTugas*. Basis data tugas siswa dapat dilihat pada Gambar 5.7.



Gambar 5.70 Implementasi Basis Data Tugas Siswa

5.2.3.4 Basis Data Uploads

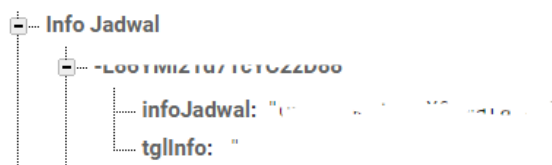
Basis data *uploads* ini digunakan untuk menyimpan data dari siswa berupa nama berkas dan url dari *file* .pdf yang diunggah ke *database*. Atribut yang terdapat pada basis data ini adalah *name* dan *url*. Basis data *uploads* dapat dilihat pada Gambar 5.71.



Gambar 5.71 Implementasi Basis Data Uploads

5.2.3.5 Basis Data Info Jadwal

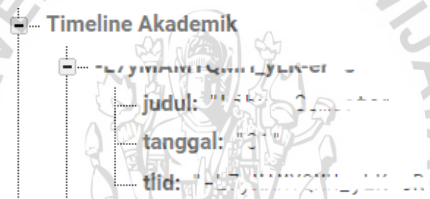
Basis data info jadwal ini digunakan untuk menyimpan data dari pengajar berupa info jadwal, serta tanggal, bulan dan tahun dari pengajar. Atribut yang terdapat pada basis data ini adalah `infoJadwal` dan `tglInfo`. Basis data info jadwal padat dilihat pada Gambar 5.72.



Gambar 5.72 Implementasi Basis Data Info Jadwal

5.2.3.6 Basis Data Timeline Akademik

Basis data *timeline* akademik ini digunakan untuk menyimpan data berupa bulan, tahun, judul dan tanggal pada menu *timeline* akademik. Atribut yang digunakan pada basis data ini adalah `judul`, `tanggal` serta `tlid` untuk menyimpan *key* dari data tersebut. Basis data *timeline* akademik dapat dilihat pada Gambar 5.73.



Gambar 5.73 Implementasi Basis Data Timeline Akademik

5.2.3.7 Basis Data Latihan Soal

Basis data latihan soal ini digunakan untuk menyimpan data berupa soal, opsi jawaban dan jawaban latihan soal. Atribut yang digunakan pada basis data ini yaitu `question`, `choice1`, `choice2`, `choice3`, `choice4`, `answer` yang masing-masing bertipe *TEXT*, serta `id` bertipe *INTEGER*. Pada basis data latihan soal hanya ditampilkan satu basis data latihan soal saja, karena untuk tabel lain memiliki struktur dan data yang sama. Untuk implementasi struktur basis data latihan soal dapat dilihat pada Gambar 5.74 dan *sample* data latihan soal dapat dilihat pada Gambar 5.75.

Latsolndo_X_soal			CREATE TABLE Latsolndo_X_soal(id INTEGER PRIMARY KEY AUTOINCREMENT,ques
id	INTEGER		`id` INTEGER PRIMARY KEY AUTOINCREMENT
question	TEXT		`question` TEXT
choice1	TEXT		`choice1` TEXT
choice2	TEXT		`choice2` TEXT
choice3	TEXT		`choice3` TEXT
choice4	TEXT		`choice4` TEXT
answer	TEXT		`answer` TEXT

Gambar 5.74 Implementasi Struktur Basis Data Latihan Soal

Table: LatsoIndo_X_soal

	id	question	choice1	choice2	choice3	choice4	answer
		Filter	Filter	Filter	Filter	Filter	Filter
1	1	Betapa apikny...	Kelebihan pen...	Cara Mangun...	Latar belakan...	Pemaparan k...	Kelebihan pen...
2	2	Terlepas dari ...	Pada bagian p...	Pada bagian p...	Pada bagian p...	Pada bagian p...	Pada bagian p...
3	3	Sulit sekali m...	Kekurangan/k...	Keunggulan/K...	Latar belakan...	Kekurangan d...	Kekurangan d...
4	4	Insiden kecil i...	Keunggulan b...	Isi pokok buku	Deskripsi buku	Kelemahan bu...	Kelemahan bu...
5	5	Cerita ini bai...	Cerita dapat d...	Karakter tidak...	Pengarang m...	Cerita diungk...	Cerita dapat d...

Gambar 5.75 Implementasi Basis Data Latihan Soal

5.2.4 Implementasi Class

Dari setiap class yang dirancang pada tahap perancangan direalisasikan pada beberapa *file* program dengan ekstensi *.java. Daftar implementasi *class* yang telah dibuat ditunjukkan pada Tabel 5.11.

Tabel 5.11 Implementasi Class Pada Kode Program Java

No	Package	Nama Class / Interface	Nama File Program
1	Aspirasi	Aspirasi	Aspirasi.java
2	Aspirasi	AdapterAspirasi	AdapterAspirasi.java
3	Aspirasi	ModelAspirasi	ModelAspirasi.java
4	JadwalPelajaran	AdapterInfo	AdapterInfo.java
5	JadwalPelajaran	JadwalPelajaranPengajar	JadwalPelajaranPengajar.java
6	JadwalPelajaran	JadwalPelajaranSiswa	JadwalPelajaranSiswa.java
7	JadwalPelajaran	Model_infoJadwal	Model_infoJadwal.java
8	JadwalPelajaran	PerubahanJadwal	PerubahanJadwal.java
9	LatihanSoal	LatihanSoalActivity	LatihanSoalActivity.java
10	LatihanSoal	Latso_keminatan_XI	Latso_keminatan_XI.java
11	LatihanSoal	Latso_keminatan_XII	Latso_keminatan_XII.java
12	LatihanSoal	Latso_keminatan_XII	Latso_keminatan_XII.java
13	LatihanSoal	Latso_IPA_XI	Latso_IPA_XI.java
14	LatihanSoal	Latso_IPS_XI	Latso_IPS_XI.java
15	LatihanSoal	Latso_BHS_XI	Latso_BHS_XI.java
16	LatihanSoal	Latso_IPA_XII	Latso_IPA_XII.java
17	LatihanSoal	Latso_IPS_XII	Latso_IPS_XII.java
18	LatihanSoal	Latso_BHS_XII	Latso_BHS_XII.java

19	LatihanSoal	Question	Question.java
20	LatihanSoal	HighestScoreActivity	HighestScoreActivity.java
21	Bio_X	LatsoBiologi_X	LatsoBiologi_X.java
22	Bio_X	LatsoBiologi_X_db	LatsoBiologi_X_db.java
23	Bio_X	LatsoBiologi_X_soal	LatsoBiologi_X_soal.java
24	Eko_X	LatsoEko_X	LatsoEko_X.java
25	Eko_X	LatsoEko_X_db	LatsoEko_X_db.java
26	Eko_X	LatsoEko_X_soal	LatsoEko_X_soal.java
27	Indo_X	LatsoIndo_X	LatsoIndo_X.java
28	Indo_X	LatsoIndo_X_db	LatsoIndo_X_db.java
29	Indo_X	LatsoIndo_X_soal	LatsoIndo_X_soal.java
30	Fis_XI	LatsoFis_XI	LatsoFis_XI.java
31	Fis_XI	LatsoFis_XI_db	LatsoFis_XI_db.java
32	Fis_XI	LatsoFis_XI_soal	LatsoFis_XI_soal.java
33	Ing_XI	LatsoIng_XI	LatsoIng_XI.java
34	Ing_XI	LatsoIng_XI_db	LatsoIng_XI_db.java
35	Ing_XI	LatsoIng_XI_soal	LatsoIng_XI_soal.java
36	Kim_XI	LatsoKim_XI	LatsoKim_XI.java
37	Kim_XI	LatsoKim_XI_db	LatsoKim_XI_db.java
38	Kim_XI	LatsoKim_XI_soal	LatsoKim_XI_soal.java
39	Mat_XII	LatsoMat_XII	LatsoMat_XII.java
40	Mat_XII	LatsoMat_XII_db	LatsoMat_XII_db.java
41	Mat_XII	LatsoMat_XII_soal	LatsoMat_XII_soal.java
42	Sej_XII	LatsoSej_XII	LatsoSej_XII.java
43	Sej_XII	LatsoSej_XII_db	LatsoSej_XII_db.java
44	Sej_XII	LatsoSej_XII_soal	LatsoSej_XII_soal.java
45	Sosio_XII	LatsoSosio_XII	LatsoSosio_XII.java
46	Sosio_XII	LatsoSosio_XII_db	LatsoSosio_XII_db.java
47	Sosio_XII	LatsoSosio_XII_soal	LatsoSosio_XII_soal.java
48	MateriPelajaran	kelas	Kelas.java
49	MateriPelajaran	kelas_x	kelas_x.java

50	MateriPelajaran	listKeminatan_xi	listKeminatan_xi.java
51	MateriPelajaran	listKeminatan_xii	listKeminatan_xii.java
52	MateriPelajaran	Kelas_ipa_xi	Kelas_ipa_xi.java
53	MateriPelajaran	Kelas_ipa_xii	Kelas_ipa_xii.java
54	MateriPelajaran	Kelas_ips_xi	Kelas_ips_xi.java
55	MateriPelajaran	Kelas_ips_xi	Kelas_ips_xi.java
56	MateriPelajaran	Kelas_bahasa_xi	Kelas_bahasa_xi.java
57	MateriPelajaran	Kelas_bahasa_xi	Kelas_bahasa_xi.java
58	Profil	ProfilSiswa	ProfilSiswa.java
59	Profil	ProfilSiswa_edit	ProfilSiswa_edit.java
60	Profil	Model_profilSiswa	Model_profilSiswa.java
61	TimelineAkademik	TimelineSiswa	TimelineSiswa.java
62	TimelineAkademik	TimelinePengajar	TimelinePengajar.java
63	TimelineAkademik	TambahKegiatan	TambahKegiatan.java
64	TimelineAkademik	AdapterTimeline	AdapterTimeline.java
65	TimelineAkademik	ModelTimeline	ModelTimeline.java
66	Tugas	AdapterTugas	AdapterTugas.java
67	Tugas	BerkasTugas	BerkasTugas.java
68	Tugas	Constant	Constant.java
69	Tugas	ModelTugas	ModelTugas.java
70	Tugas	ModelUpload	ModelUpload.java
71	Tugas	NotificationGenerator	NotificationGenerator.java
72	Tugas	TugasPengajar	TugasPengajar.java
73	Tugas	TugasSiswa	TugasSiswa.java
74	Tugas	UngahTugas	UngahTugas.java
75	UserAuth	DB_AkunPengajar	DB_AkunPengajar.java
76	UserAuth	LoginPengajar	LoginPengajar.java
77	UserAuth	LoginSiswa	LoginSiswa.java
78	UserAuth	LogonScreen	LogonScreen.java
79	UserAuth	Pendaftaran	Pendaftaran.java
80	UserAuth	RegisterPengajar	RegisterPengajar.java

81	UserAuth	RegisterSiswa	RegisterSiswa.java
82		CustomGridViewActivity	CustomGridViewActivity.java
83		MenuPengajar	MenuPengajar.java
84		MenuSiswa	MenuSiswa.java

5.2.5 Implementasi Kode Program

Bagian bagian ini akan menjelaskan tentang implementasi kode program utapa pada aplikasi E-Learning SMANTIGDA dengan menggunakan bahasa Java.

5.2.5.1 Implementasi Method tambahTugas()

Method tambahTugas digunakan oleh sistem untuk memberikan tugas yang diberikan oleh pengajar kepada siswa dan menampilkan daftar tugas yang telah diberikan. Kode program *method* tambahTugas ditunjukkan pada Gambar 5.76.

```

1.  public void tambahTugas() {
2.      String Matpel = matpel
3.      .getSelectedItem().toString();
4.      String Tema = tema.getText().toString().trim();
5.      String Index = index.getText().toString().trim();
6.      String Kelas = kelas
7.      .getSelectedItem().toString().trim();
8.      String Jam = jam
9.      .getSelectedItem().toString().trim();
10.     String Meridiem = meridiem
11.     .getSelectedItem().toString().trim();
12.     String Tanggal = tanggal
13.     .getSelectedItem().toString().trim();
14.     String Bulan = bulan
15.     .getSelectedItem().toString().trim();
16.     String Tahun = tahun
17.     .getSelectedItem().toString().trim();
18.     String Deskripsi = deskripsi
19.     .getText().toString().trim();
20.     if (!TextUtils.isEmpty(Tema)) {
21.         String id = dbTugas.push().getKey();
22.         ModelTugas tugas = new ModelTugas(id, Matpel,
23.         Tema, Index, Kelas, Jam, Meridiem,
24.         Tanggal, Bulan, Tahun,
25.         Deskripsi);
26.         dbTugas.child(id).setValue(tugas);
27.         tema.setText("");
28.         index.setText("");
29.         deskripsi.setText("");
30.         Toast.makeText(this, "Tugas telah
31.         ditambahkan",
32.         Toast.LENGTH_LONG).show();
33.     } else {
34.         Toast.makeText(this, "Masukkan data terlebih
35.         dahulu", Toast.LENGTH_LONG).show();
36.     }
37. }

```

Gambar 5.76 Implementasi *Method* tambahTugas()

Penjelasan kode program *method* tambahTugas ditunjukkan pada Tabel 5.12 dibawah.

Tabel 5.12 Penjelasan *Method* tambahTugas()

Nomor Kode Program	Penjelasan
1	Inisialisasi metode tambahTugas
3-21	Deklarasi objek untuk melakukan pengambilan data masukan dari pengguna dan merubahnya menjadi string.
23	Kondisi jika variabel Tema tidak kosong, akan melakukan inisialisasi variabel id bertipe String untuk membuat id pada database. Kemudian melakukan inisialisai objek tugas yang mereference pada ModelTugas. Mengirimkan value objek tugas pada variabel dbTugas, mengubah beberapa teks pada variabel menjadi kosong dan menampilkan teks jika proses berhasil.
36	Jika tidak melakukan tugas, maka akan menampilkan teks.

5.2.5.2 Implementasi Method uploadFile()

Method uploadFile digunakan oleh sistem agar siswa dapat melakukan unggah berkas tugas pada *database* agar selanjutnya dapat dilihat oleh pengajar. Kode program *method* uploadFile ditunjukkan oleh Gambar 5.77.

1.	<code>private void uploadFile(Uri data) {</code>
2.	<code> progressBar.setVisibility(View.VISIBLE);</code>
3.	<code> StorageReference sRef = mStorageReference</code>
4.	<code> .child(Constant.STORAGE_PATH_UPLOADS + System</code>
5.	<code> .currentTimeMillis() + ".pdf");</code>
6.	<code> sRef.putFile(data).addOnSuccessListener(new</code>
7.	<code> OnSuccessListener<UploadTask.TaskSnapshot>() {</code>
8.	<code> @SuppressWarnings("VisibleForTests")</code>
9.	<code> @Override</code>
10.	<code> public void onSuccess(UploadTask.TaskSnapshot</code>
11.	<code> taskSnapshot) {</code>
12.	<code> progressBar.setVisibility(View.GONE);</code>
13.	<code> textViewStatus.setText("Tugas berhasil</code>
14.	<code> diunggah");</code>
15.	<code> ModelUpload upload = new ModelUpload</code>
16.	<code> (editTextFilename.getText().toString(),</code>
17.	<code> taskSnapshot.getDownloadUrl().toString());</code>
18.	<code> </code>
19.	<code> mDatabaseReference</code>
20.	<code> .child(mDatabaseReference</code>
21.	<code> .push().getKey()).setValue(upload);</code>
22.	<code> }</code>
23.	<code> })</code>
24.	<code> }.addOnFailureListener(new OnFailureListener() {</code>
25.	<code> @Override</code>
26.	<code> public void onFailure(@NonNull Exception</code>
27.	<code> exception){</code>
28.	<code> Toast.makeText(getApplicationContext(),</code>
29.	<code> exception.getMessage(), Toast.LENGTH_LONG)</code>
30.	<code> .show();</code>

```

31.         }
32.     })
33.     .addOnProgressListener(new OnProgressListener
34.     <UploadTask.TaskSnapshot>() {
35.         @SuppressWarnings("VisibleForTests")
36.         @Override
37.         public void
38.         onProgress(UploadTask.TaskSnapshot
39.         taskSnapshot) {
40.             double progress = (100.0 *
41.             taskSnapshot.getBytesTransferred()) /
42.             taskSnapshot.getTotalByteCount();
43.             textViewStatus.setText((int) progress
44.             + "% Uploading...");
45.         }
46.     });
47. }

```

Gambar 5.77 Implementasi Method uploadFile()

Penjelasan kode program *method* uploadFile ditunjukkan pada Tabel 5.13.

Tabel 5.13 Penjelasan Method uploadFile()

Nomor Kode Program	Penjelasan
1-6	Inisialisasi metode uploadFile dengan data bertipe Uri, dengan melakukan deklarasi variabel sRef bertipe StorageReference yang merujuk pada child di <i>database</i> .
10-23	Inisialisasi metode addOnSuccessListener. Terdapat kelas onSuccess melakukan aksi menampilkan teks "Tugas Berhasil Diunggah" jika berhasil. Serta mendapatkan <i>value</i> teks nama berkas dan teks url dari kelas ModelUpload dan mengubahnya menjadi bentuk <i>string</i> . Kemudian melakukan setValue pada variabel <i>upload</i> ke database dengan menyertakan id file.
24-32	Inisialisasi addOnFailureListener, melakukan tampilkan pesan gagal jika tidak dapat melakukan aktivitas unggah.
33	Inisialisasi metode addOnProgressListener, menampilkan progress saat melakukan aktivitas unggah.

5.2.5.3 Implementasi Method addAspirasi()

Method addAspirasi digunakan oleh sistem agar siswa dan pengajar agar dapat mengirimkan aspirasinya, kemudian semua aspirasi yang dikirimkan dapat dilihat oleh seluruh pengguna aplikasi. Kode program *method* addAspirasi ditunjukkan oleh Gambar 5.78.

```

1.     private void addAspirasi() {
2.         String isiAspirasi = txtAspirasi.getText()
3.         .toString().trim();
4.         String sasaran = spinSasaranAspirasi
5.         .getSelectedItem().toString();

```

6.	
7.	<code>if (!TextUtils.isEmpty(isiAspirasi)) {</code>
8.	<code>String id = databaseAspirasi.push().getKey();</code>
9.	<code>ModelAspirasi aspirasi = new ModelAspirasi(id,</code>
10.	<code>isiAspirasi, sasaran);</code>
11.	
12.	<code>databaseAspirasi.child(id).setValue(aspirasi);</code>
13.	<code>txtAspirasi.setText("");</code>
14.	<code>Toast.makeText(this, "Aspirasi telah</code>
15.	<code>ditambahkan", Toast.LENGTH_LONG).show();</code>
16.	<code>} else {</code>
17.	<code>Toast.makeText(this, "Masukkan aspirasi</code>
18.	<code>terlebih dahulu", Toast.LENGTH_LONG).show();</code>
19.	<code>}</code>
20.	<code>}</code>

Gambar 5.78 Implementasi Method addAspirasi()

Penjelasan kode program *method* addAspirasi ditunjukkan pada Tabel 5.14.

Tabel 5.14 Penjelasan Method addAspirasi()

Nomor Kode Program	Penjelasan
1	Inisialisasi metode dengan nama addAspirasi()
2-5	Inisialisasi variabel isiAspirasi dan sasaran bertipe String untuk mengambil data yang telah diberikan oleh pengguna
7-15	Pengkodisian jika <i>text value</i> tidak kosong, maka akan melakukan deklarasi <i>key id</i> aspirasi dan mengirimkan data melalui kelas ModelAspirasi. Data akan dikirim menggunakan variabel databaseAspirasi pada child aspirasi. Setelah selesai dikirim <i>text value</i> akan dibuat kosong kembali dan sistem akan melakukan <i>toasting text</i> dengan <i>value</i> "Aspirasi telah ditambahkan"
16-19	Pengkondisian jika tidak ada <i>text</i> maka akan melakukan <i>toast-ing text</i> dengan <i>value</i> "Masukkkkan aspiari terlebih dahulu"

5.2.6 Implementasi Antarmuka

Hasil dari implementasi antarmuka aplikasi E-Learning SMANTIGDA akan dijelaskan pada tahapan ini. Pada tahap ini akan dijelaskan tentang *screenflow* antarmuka dan detil dari tiap antarmuka.

5.2.6.1 Scerenflow Antarmuka

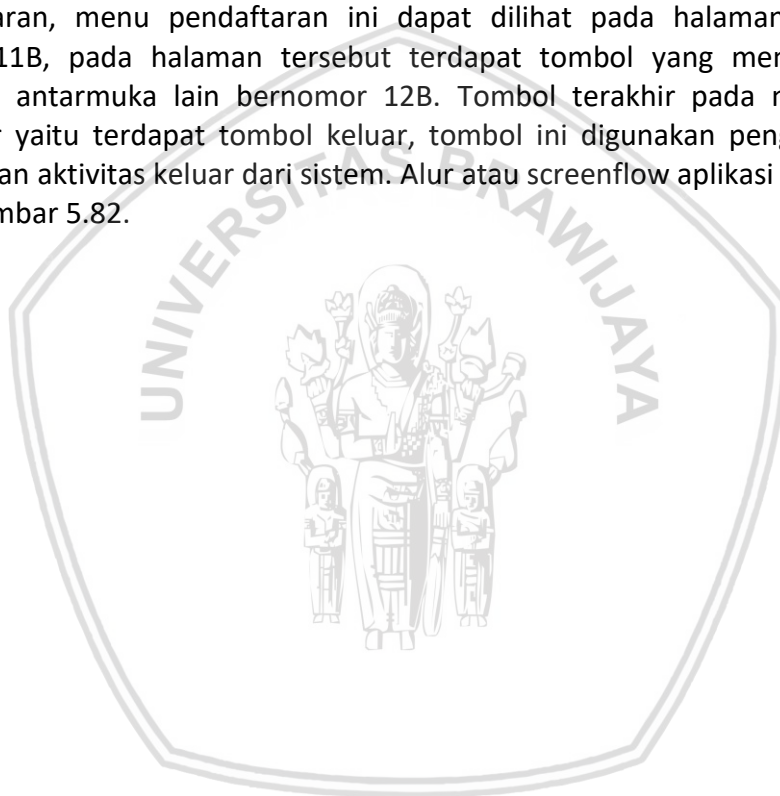
Alur dari antarmuka diawali dari nomor 1 yang menunjukkan halaman *login screen*, pada halaman tersebut terdapat dua buat tombol yang menghubungkan halaman *login* siswa dan pengajar yang ditunjukkan oleh nomor 2A sebagai halaman antarmuka *login* siswa dan 2B sebagai halaman antarmuka *login*

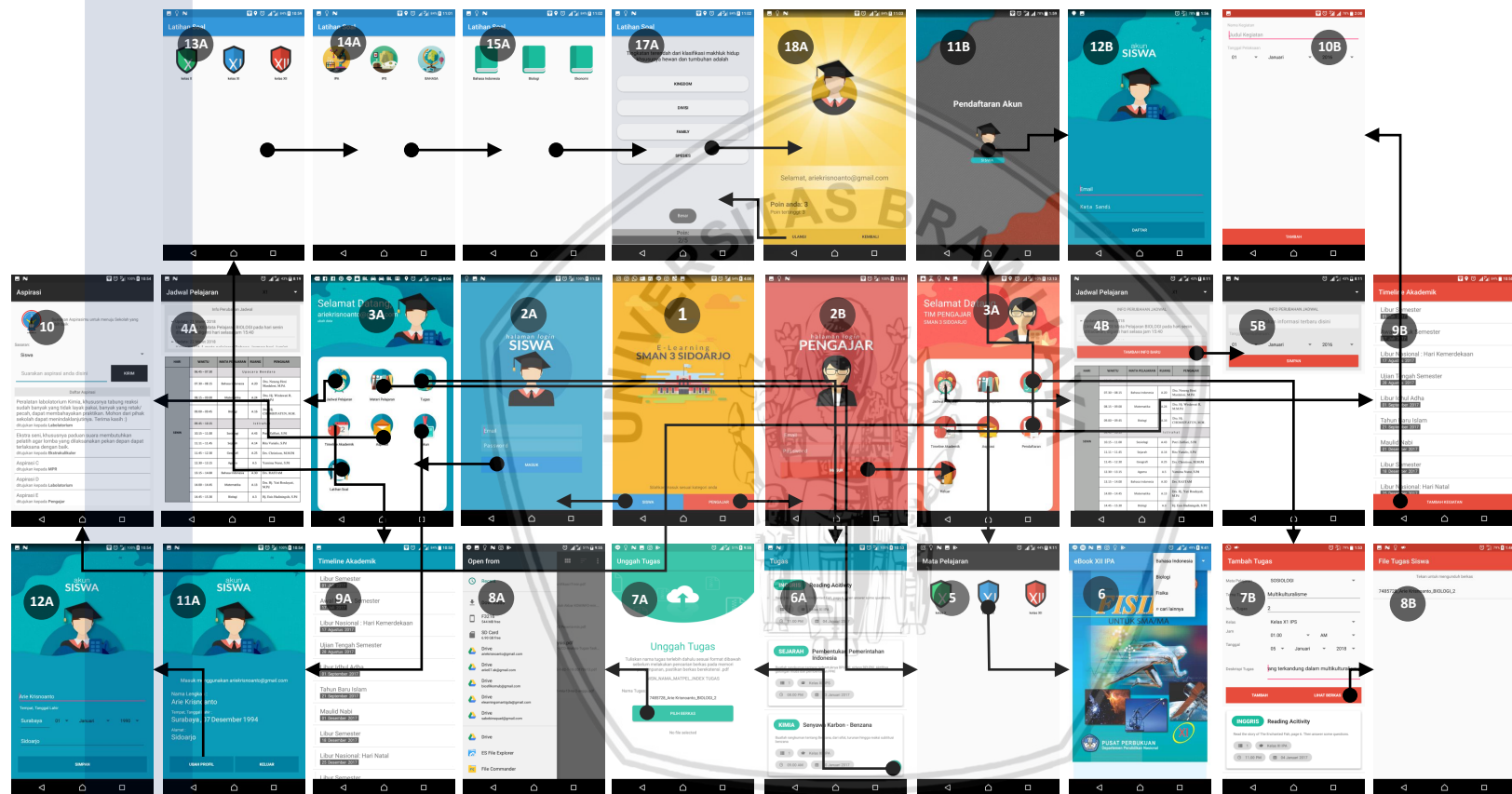
pengajar. Dari halaman antarmuka *login* siswa dan *login* pengajar, alur selanjutnya setelah menekan tombol masuk pada masing-masing halaman *login* yaitu masuk ke halaman menu utama siswa yang ditunjukkan pada nomor 3A dan halaman menu utama pengajar yang ditunjukkan pada nomor 3B.

Dari halaman antarmuka menu utama siswa, terdapat tujuh tombol menu. Menu pertama yaitu jadwal pelajaran, jika tombol menu jadwal pelajaran ditekan, sistem akan mengarahkan pada halaman jadwal pelajaran siswa yang ditunjukkan pada nomor 4A. Menu kedua yaitu materi pelajaran, jika tombol menu materi pelajaran ditekan sistem akan mengarah pada halaman materi pelajaran yang ditunjukkan pada nomor 5, pengguna akan diarahkan untuk memilih beberapa pilihan hingga menuju ke halaman antarmuka latihan soal yang menampilkan dokumen materi pelajaran seperti yang ditunjukkan pada halaman antarmuka nomor 6. Menu ketiga adalah menu tugas, menu tugas jika ditekan akan mengarah pada halaman antarmuka tugas siswa yang ditunjukkan oleh nomor 6A, pada halaman tersebut terdapat sebuah tombol dikiri bawah untuk mengarahkan pada halaman unggah tugas yang ditunjukkan oleh nomor 6A, pada halaman unggah tugas juga terdapat tombol lagi untuk menuju kehalaman *file manager* untuk melakukan pencarian berkas tugas yang akan diunggah, laman ini ditunjukkan pada nomor 8A. Menu selanjutnya pada sisi siswa yaitu menu *timeline* akademik yang ditunjukkan nomor 9A, setelah itu tombol sebelah kanannya terdapat menu aspirasi yang akan mengarah pada halaman antarmuka yang ditunjukkan oleh nomor 10. Beralih pada menu selanjutnya terdapat menu akun, menu ini akan mengarah pada halaman antarmuka yang ditunjukkan pada nomor 11A, pada bagian bawah halaman terdapat dua buah tombol, keluar dan ubah profil, jika pengguna menekan tombol ubah profil, pengguna akan diarahkan pada halaman antarmuka yang ditunjukkan pada nomor 12A. Tombol terakhir pada menu utama siswa yaitu tombol menu latihan soal, tombol menu latihan soal jika ditekan akan mengarah pada halaman antarmuka yang ditunjukkan pada nomor 13A, pada halaman ini pengguna diarahkan untuk memilih kelas, setelah memilih pengguna akan diarahkan pada halaman antarmuka nomor 14A untuk memilih keminata mata pelajaran, setelah memilih selanjutnya pengguna akan diarahkan menuju halaman antarmuka nomor 15A, lalu pengguna akan diarahkan ke halaman antarmuka latihan soal yang ditunjukkan pada halaman antarmuka nomor 17A, setelah selesai mengerjakan latihan soal pengguna akan diarahkan ke halaman antarmuka perolehan skor latihan soal seperti yang ditunjukkan pada nomor 18A.

Pada sisi pengajar, menu pertama pada menu pengajar yaitu terdapat tombol jadwal pelajaran, tombol ini mengarah pada halaman antarmuka nomor 4B, pada halaman antarmuka ini terdapat tombol lain yang mengarah pada halaman fitur tambah info perubahan jadwal yang dapat dilihat pada halaman antarmuka nomor 5B. Menu selanjutnya yaitu menu materi pelajaran, menu ini menuju ke halaman yang sama dengan pengguna siswa, menu ini mengarah pada halaman antarmuka nomor 5, yang kemudian pengguna akan diarahkan menuju ke halaman antarmuka materi pelajaran yang menampilkan dokumen materi

seperti pada halaman antarmuka nomor 6. Menu selanjutnya terdapat menu tugas pengajar, menu ini ditunjukkan pada halaman antarmuka yang dapat dilihat pada nomor 7B, pada halaman antarmuka ini juga terdapat tombol lihat berkas yang mengarah pada halaman antarmuka lain yang dapat dilihat pada halaman antarmuka nomor 8B. Setelah menu tugas pengajar, terdapat menu *timeline* akademik yang dapat dilihat pada halaman antarmuka nomor 9B, halaman antarmuka pada fitur ini terdapat tombol pada bagian bawah halaman yang akan mengarah pada halaman lain yang dapat dilihat pada nomor 10B yang berfungsi untuk menambah daftar kegiatan. Menu selanjutnya terdapat menu aspirasi yang ditunjukkan dengan halaman antarmuka bernomor 10, halaman ini merupakan halaman yang sama seperti halaman yang diakses oleh pengguna siswa. Menu terakhir yang mengarah pada halaman antarmuka lain yaitu menu pendaftaran, menu pendaftaran ini dapat dilihat pada halaman antarmuka nomor 11B, pada halaman tersebut terdapat tombol yang mengarah pada halaman antarmuka lain bernomor 12B. Tombol terakhir pada menu utama pengajar yaitu terdapat tombol keluar, tombol ini digunakan pengguna untuk melakukan aktivitas keluar dari sistem. Alur atau screenflow aplikasi dapat dilihat pada gambar 5.82.





Gambar 5.79 Screenflow Antarmuka

5.2.6.2 Detil Implementasi Antarmuka

Implementasi antarmuka aplikasi ini memiliki tampilan yang hampir sama dari kedua sisi pengguna, yaitu sisi siswa dan sisi pengajar, namun dibedakan dengan warna dasar tiap *layout*-nya

(a) Halaman Logon Screen

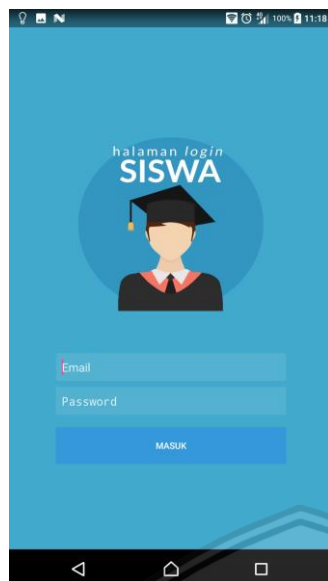
Halaman *logon* merupakan halaman paling awal yang muncul jika baru pertama kali membuka aplikasi ini. Pada halaman ini akan diperlihatkan dua tombol untuk melakukan login sebagai siswa atau login sebagai pengajar. Antarmuka halaman *logon* ditunjukkan oleh Gambar 5.80.



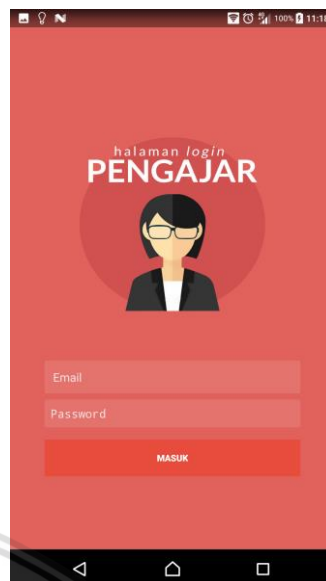
Gambar 5.80 Implementasi Antarmuka Logon Screen

(b) Halaman Login

Halaman *login* merupakan halaman yang digunakan oleh pengguna untuk masuk ke dalam aplikasi. Pengguna dapat memasukkan data *email* dan *password*. Antarmuka halaman *login* siswa dapat dilihat pada Gambar 5.81 dan pengajar dapat dilihat pada Gambar 5.82.



Gambar 5.81 Implementasi Antarmuka Login Siswa



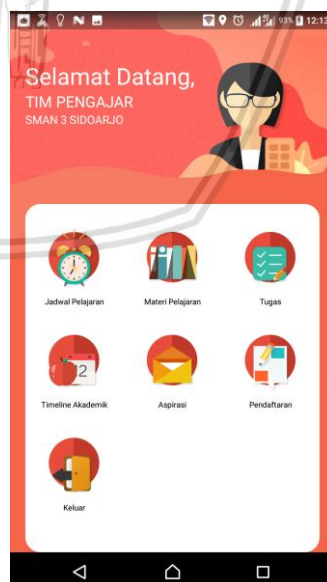
Gambar 5.82 Implementasi Antarmuka Login Pengajar

(c) Halaman Menu Utama

Hasil halaman antarmuka menu siswa dan menu pengajar, terdapat perbedaan warna dasar untuk membedakan, menu siswa memiliki warna dasar *teal*, menu pengejar memiliki warna dasar *cinnabar*. Untuk halaman menu utama siswa dapat dilihat pada Gambar 5.83 dan halaman menu utama pengajar dapat dilihat pada Gambar 5.84.



Gambar 5.83 Implementasi Antarmuka Halaman Utama Siswa



Gambar 5.84 Implementasi Antarmuka Halaman Utama Pengajar

(d) *Halaman Jadwal Pelajaran*

Halaman jadwal pelajaran ini menampilkan jadwal pelajaran yang dinavigasi oleh *spinner* pada bagian kanan atas untuk memilih kelas, *spinner* tersebut akan menampilkan pilihan kelas untuk dipilih oleh pengguna. Yang membedakan antara kedua antarmuka dari siswa dan pengajar adalah terdapat *button* tambah info baru pada laman jadwal pelajaran pengajar.

Jadwal Pelajaran

X1

Info Perubahan Jadwal

➤ Update: 21 Maret 2018

Lintuk kelas XI Mata Pelajaran BIOLOGI pada hari senin diadipindah, diganti hari selasa jam 15:40

➤ Update: 22 Maret 2018

Mata XI IPA 3 mata pelajaran Bahasa Indonesia, Fisika

HARI	WAKTU	MATA PELAJARAN	RUANG	PENGAJAR
SENIN	06.45 - 07.30	Upacara Bendera		
	07.30 - 08.15	Bahasa Indonesia	A.20	Drs. Neneng Hesti Maridani, M.Pd
	08.15 - 09.00	Matematika	A.24	Drs. H. Widiawati R. M.M.Pd
	09.00 - 09.45	Biologi	A.16	Drs. H. CHOMSIYATUN, M.M.
	09.45 - 10.15	Istirahat		
	10.15 - 11.00	Sosiologi	A.43	Purni Zulfati, S.Pd
	11.11 - 11.45	Sejarah	A.14	Rita Variati, S.Pd
	11.45 - 12.30	Geografi	A.25	Drs. Christion, M.M.Pd
	12.30 - 13.15	Agama	A.5	Yunima Nurul, S.Pd
	13.15 - 14.00	Bahasa Indonesia	A.30	Drs. RASTAM
	14.00 - 14.45	Matematika	A.13	Drs. H. Yati Rosdayati, M.Pd
	14.45 - 15.30	Biologi	A.3	Hj. Eain Haidingsih, S.Pd

Gambar 5.85 Implementasi Antarmuka Halaman Jadwal Pelajaran Siswa

Jadwal Pelajaran

Info Perubahan Jadwal

- Update: 21 Maret 2018
Untuk kelas X3 Mata Pelajaran BIOLOGI pada
diadakan, diganti hari selasa jam 15:40
- Update: 22 Maret 2018
Mata Pelajaran X1

HARI	WAKTU	MATA PELAJARAN	RUANG	PENGAJAR	
SENIN	06.45 - 07.30	Upacara Bendera			
	07.30 - 08.15	Bahasa Indonesia	A.20	Drs. Neneng Hesti Maridani, M.Pd	
	08.15 - 09.00	Matematika	A.24	Drs. H. Widiawati R. M.M.Pd	
	09.00 - 09.45	Biologi	A.16	Drs. H. CHOMSIYATUN, M.M.	
	09.45 - 10.15	Istirahat			
	10.15 - 11.00	Sosiologi	A.43	Purni Zulfati, S.Pd	
	11.11 - 11.45	Sejarah	A.14	Rita Variati, S.Pd	
	11.45 - 12.30	Geografi	A.25	Drs. Christion, M.M.Pd	
	12.30 - 13.15	Agama	A.5	Yunima Nurul, S.Pd	
	13.15 - 14.00	Bahasa Indonesia	A.30	Drs. RASTAM	
	14.00 - 14.45	Matematika	A.13	Drs. H. Yati Rosdayati, M.Pd	
	14.45 - 15.30	Biologi	A.3	Hj. Eain Haidingsih, S.Pd	

Gambar 5.86 Implementasi Antarmuka Halaman Jadwal Pelajaran Siswa 2

Jadwal Pelajaran

INFO PERUBAHAN JADWAL

➤ Update 21 Maret 2018

Untuk kelas XI Mata Pelajaran BIOLOGI pada hari senin ditidurkan, diganti hari selasa jam 15:40

TAMBAH INFO BARU

HARI	WAKTU	MATA PELAJARAN	RUANG	PENGAJAR
SENIN	06.45 - 07.30	Upacara Bendera		
	07.30 - 08.15	Bahasa Indonesia	A.20	Drs. Neneng Hesti Maridani, M.Pd
	08.15 - 09.00	Matematika	A.24	Drs. H. Widiawati R. M.M.Pd
	09.00 - 09.45	Biologi	A.16	Drs. H. CHOMSIYATUN, M.M.
	09.45 - 10.15	Istirahat		
	10.15 - 11.00	Sosiologi	A.43	Purni Zulfati, S.Pd
	11.11 - 11.45	Sejarah	A.14	Rita Variati, S.Pd
	11.45 - 12.30	Geografi	A.25	Drs. Christion, M.M.Pd
	12.30 - 13.15	Agama	A.5	Yunima Nurul, S.Pd
	13.15 - 14.00	Bahasa Indonesia	A.30	Drs. RASTAM
	14.00 - 14.45	Matematika	A.13	Drs. H. Yati Rosdayati, M.Pd
	14.45 - 15.30	Biologi	A.3	Hj. Eain Haidingsih, S.Pd

Gambar 5.88 Implementasi Antarmuka Halaman Jadwal Pelajaran Pengajar

Jadwal Pelajaran

INFO PERUBAHAN JADWAL

➤ Update: 21 Maret 2018

Untuk kelas XI Mata Pelajaran BIOLOGI pada
diadakan, diganti hari selasa jam 15:40

TAMBAH INFO BARU

X1

X2

XI IPA

XI IPS

XI Bahasa

XII IPA

XII IPS

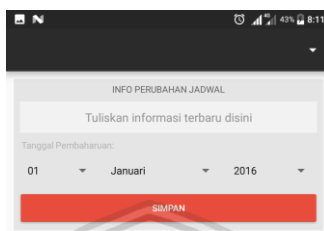
XII Bahasa

HARI	WAKTU	MATA PELAJARAN	RUANG	PENGAJAR
SENIN	06.45 - 07.30	Upacara Bendera		
	07.30 - 08.15	Bahasa Indonesia	A.20	Drs. Neneng Hesti Maridani, M.Pd
	08.15 - 09.00	Matematika	A.24	Drs. H. Widiawati R. M.M.Pd
	09.00 - 09.45	Biologi	A.16	Drs. H. CHOMSIYATUN, M.M.
	09.45 - 10.15	Istirahat		
	10.15 - 11.00	Sosiologi	A.43	Purni Zulfati, S.Pd
	11.11 - 11.45	Sejarah	A.14	Rita Variati, S.Pd
	11.45 - 12.30	Geografi	A.25	Drs. Christion, M.M.Pd
	12.30 - 13.15	Agama	A.5	Yunima Nurul, S.Pd
	13.15 - 14.00	Bahasa Indonesia	A.30	Drs. RASTAM
	14.00 - 14.45	Matematika	A.13	Drs. H. Yati Rosdayati, M.Pd
	14.45 - 15.30	Biologi	A.3	Hj. Eain Haidingsih, S.Pd

Gambar 5.87 Implementasi Antarmuka Halaman Jadwal Pelajaran Pengajar 2

(e) Halaman Tambah Info Perubahan Jadwal

Halaman tambah info perubahan jadwal ini terdapat pada sisi pengajar. Pada halaman ini terdapat *form* untuk memasukkan data teks berupa info perubahan jadwalnya, kemudian terdapat tiga *spinner* untuk memasukkan tanggal pembaharuan info perubahan jadwalnya. Juga terdapat tombol simpan berwarna merah untuk mengirimkan data perubahan jadwalnya. Halaman tambah info perubahan jadwal dapat dilihat pada Gambar 5.89.



Gambar 5.89 Implementasi Antarmuka Tambah Info Perubahan Jadwal

(f) Halaman Materi Pelajaran

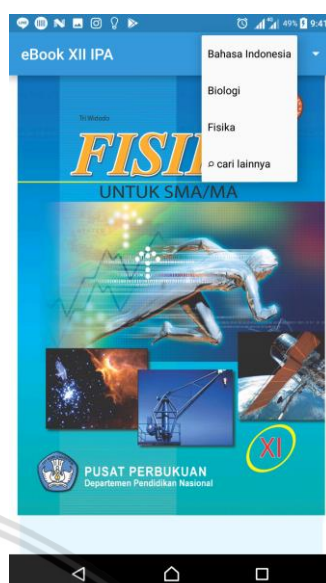
Halaman materi pelajaran ini digunakan untuk menampilkan dokumen *.pdf *ebook* penunjang berdasarkan kategori yang dipilih oleh pengguna, kategori dibagi berdasarkan kelas dan keminatannya. Serta terdapat *spinner* di pojok kanan atas untuk memilih materi pelajaran. Halaman materi pelajaran dapat dilihat pada Gambar 5.90 dan 5.91.

(g) Halaman Tugas Pengajar

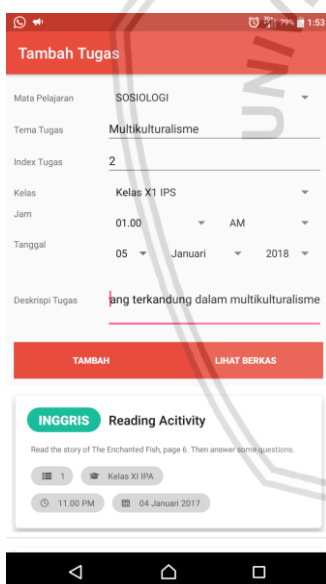
Halaman tambah tugas pengajar terdapat pada menu pengajar, pada laman ini pengajar dapat menambahkan tugas yang akan diberikan pada siswa dengan memasukkan data seperti mata pelajaran, tema pelajaran, index tugas, tujuan kelas, *deadline* tugas berupa jam, tanggal, bulan, dan tahun, dan deskripsi tugas. Halaman tugas pengajar dan perubahannya dapat dilihat pada Gambar 5.92, 5.93 dan 5.94.



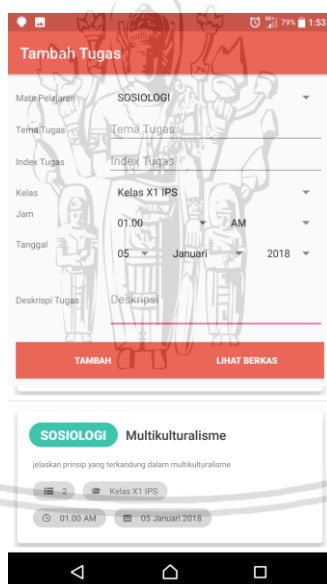
Gambar 5.91 Implementasi Antarmuka Halaman Mata Pelajaran



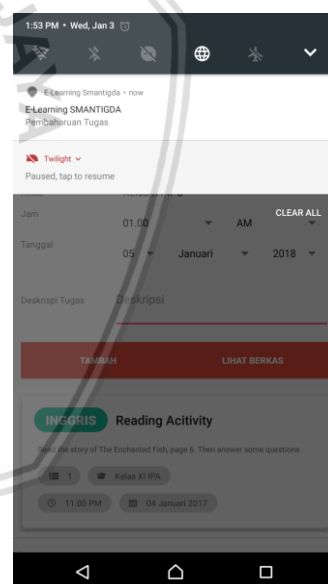
Gambar 5.90 Implementasi Antarmuka Halaman Mata Pelajaran 2



Gambar 5.93 Implementasi Antarmuka Halaman Pengajar



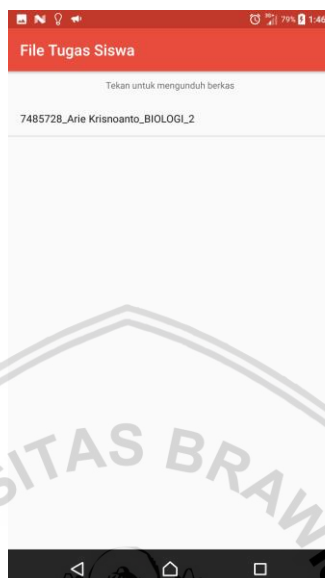
Gambar 5.92 Implementasi Antarmuka Halaman Pengajar 2



Gambar 5.94 Implementasi Antarmuka Halaman Tugas Pengajar (notifikasi)

(h) Halaman Lihat Berkas

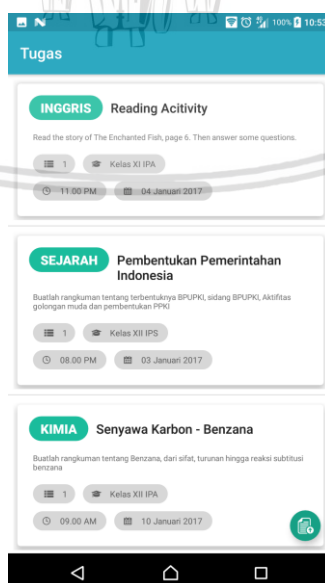
Halaman lihat berkas ini berada pada sisi pengajar, halaman ini digunakan untuk menampilkan daftar berkas tugas dari siswa yang telah dikirimkan ke *database* untuk pengajar. Halaman lihat berkas dapat dilihat pada Gambar 5.95.



Gambar 5.95 Implementasi Antarmuka Lihat Berkas

(i) Halaman Tugas Siswa

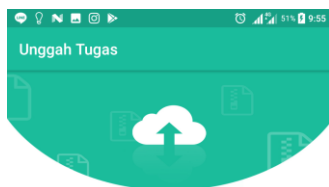
Halaman tugas siswa ini berada pada sisi siswa, halaman ini digunakan untuk menampilkan daftar tugas beserta detail tugas yang diberikan pengajar sebelumnya. Pada pojok kanan bawah terdapat tombol untuk menuju ke halaman unggah tugas. Halaman tugas siswa ditunjukkan pada Gambar 5.96.



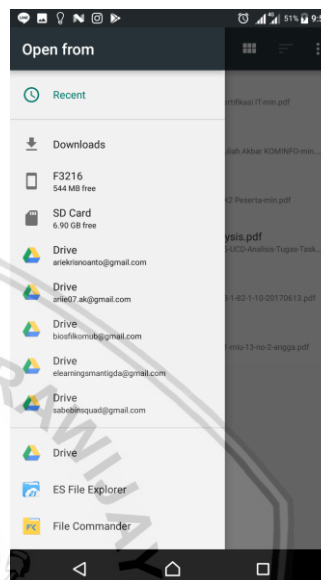
Gambar 5.96 Implementasi Antarmuka Tugas Siswa

(j) Halaman Unggah Tugas

Halaman unggah tugas ini berada pada sisi siswa, halaman ini terdapat *form* untuk memasukkan data nama berkas dan tombol untuk memilih berkas tugas yang akan diunggah. Halaman unggah tugas dapat dilihat pada Gambar 5.97. Pada gambar kedua terdapat *file manager* Android untuk memilih berkas tugas yang akan diunggah, tampilan seperti pada Gambar 5.98.



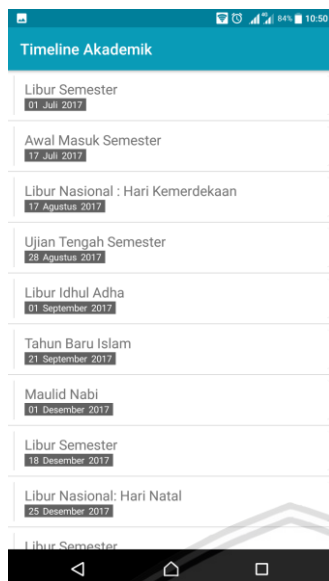
Gambar 5.97 Implementasi Antarmuka Halaman Unggah Tugas



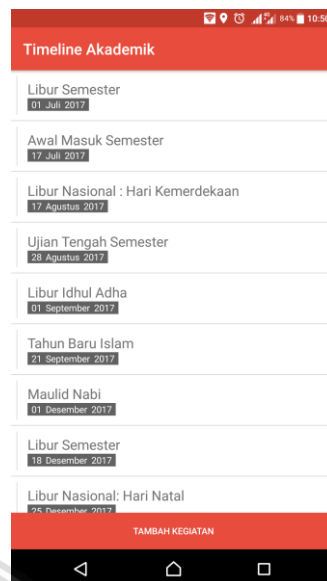
Gambar 5.98 Implementasi Antarmuka Halaman Unggah Tugas 2

(k) Halaman Timeline Akademik

Halaman antarmuka timeline akademik ini memiliki perbedaan antara siswa dan pengajar. Pada laman siswa terdapat tanggal kegiatan dan nama kegiatan selama satu periode tahun ajaran, pada sisi pengajar ada tambahan tombol tambah kegiatan untuk menambah kegiatan akademik khusus dari sekolah. Halaman timeline akademik siswa ditunjukkan pada Gambar 5.99 dan untuk pengajar ditunjukkan pada Gambar 5.100.



Gambar 5.99 Implementasi Antarmuka Timeline Akademik (Siswa)



Gambar 5.100 Implementasi Antarmuka Timeline Akademik (Pengajar)

(I) Halaman Tambah Kegiatan Akademik

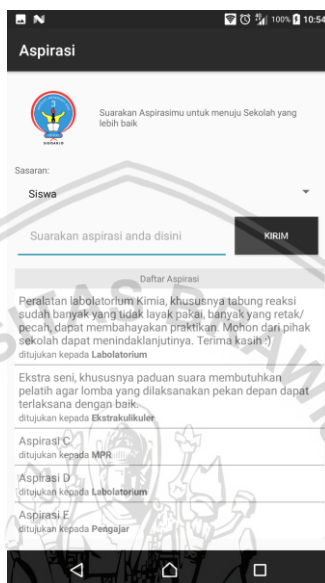
Halaman tambah kegiatan akademik ini hanya ada pada sisi pengajar saja. Pada laman ini terdapat *form* untuk masukan data informasi judul kegiatan berupa teks, kemudian dibawahnya terdapat tiga buah spinner untuk memasukkan tanggal, bulan dan tahun kegiatan akademik dari sekolah akan dilaksanakan. Halaman tambah kegiatan akademik ditunjukkan pada Gambar 5.101.



Gambar 5.101 Implementasi Antarmuka Tambah Kegiatan Akademik

(m) Halaman Aspirasi

Halaman aspirasi ini digunakan untuk menyuarakan aspirasi siswa dan pengajar yang ditujukan kepada beberapa elemen sekolah seperti laboratorium, ekstrakurikuler, dan lain sebagainya. Pada laman ini terdapat masukan data berupa teks untuk menuliskan aspirasinya, terdapat *spinner* untuk memulih tujuan aspirasinya dan tombol kirim untuk mengirimkan aspirasinya ke *database* serta ditambahkan pada bagian bawah halaman. Halaman aspirasi ditunjukkan pada Gambar 5.102.



Gambar 5.102 Implementasi Antarmuka Aspirasi

(n) Halaman Akun

Halaman akun ini menampilkan informasi akun yang sedang digunakan pada sistem, informasi yang ditampilkan berupa *email* pengguna, tempat dan tanggal lahir pengguna dan asal pengguna. Pada bagian bawah halaman juga terdapat dua buah tombol yaitu ubah profil untuk menuju ke halaman ubah profil dan tombol keluar untuk keluar dari sistem. Halaman akun ditunjukkan pada Gambar 5.103.



Gambar 5.103 Implementasi Antarmuka Akun

(o) Halaman Ubah Profil

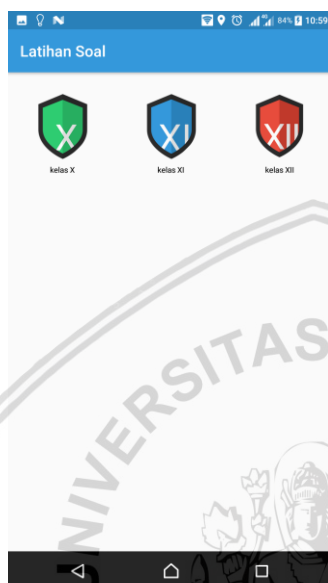
Halaman ubah profil ini terdapat tiga buah *form* yang digunakan untuk memasukkan data nama pengguna, tempat lahir pengguna dan asal pengguna. Kemudian juga terdapat tiga buah *spinner* untuk mengganti tanggal, bulan dan tahun lahir pengguna. Pada bagian bawah halaman juga terdapat tombol simpan untuk menyimpan perubahan data pengguna. Halaman ubah profil ditunjukkan pada Gambar 5.104.



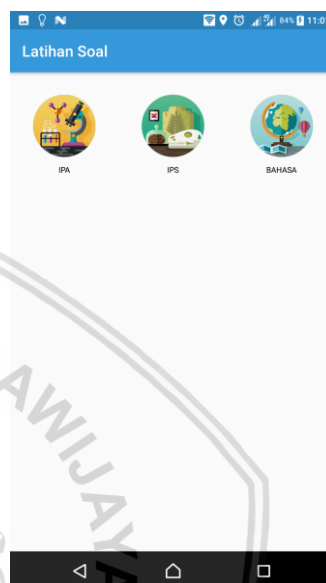
Gambar 5.104 Implementasi Antarmuka Ubah Akun

(p) Halaman Latihan Soal

Halaman latihan soal ini terdapat beberapa macam tampilan, diawal saat masuk menu latihan soal, pengguna diminta untuk memilih kelas, keminatan dan mata pelajaran. Pada laman latihan soal, bagian atas terdapat soal yang ditampilkan, kemudian juga terdapat empat buah tombol yang berisi opsi jawaban, serta bagian bawah laman terdapat perolehan poin sementara. Halaman latihan soal ditunjukkan pada Gambar 5.105, 5.106, 5.107 dan 5.108.



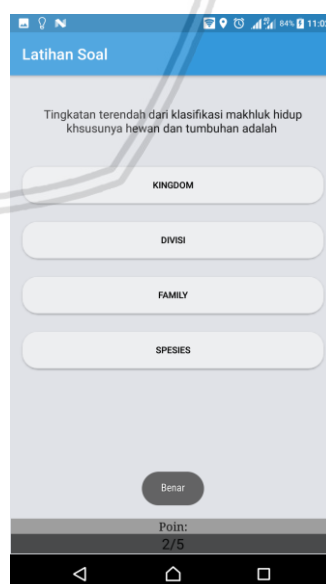
Gambar 5.105 Implementasi Antarmuka Latihan Soal 1



Gambar 5.106 Implementasi Antarmuka Latihan Soal 2



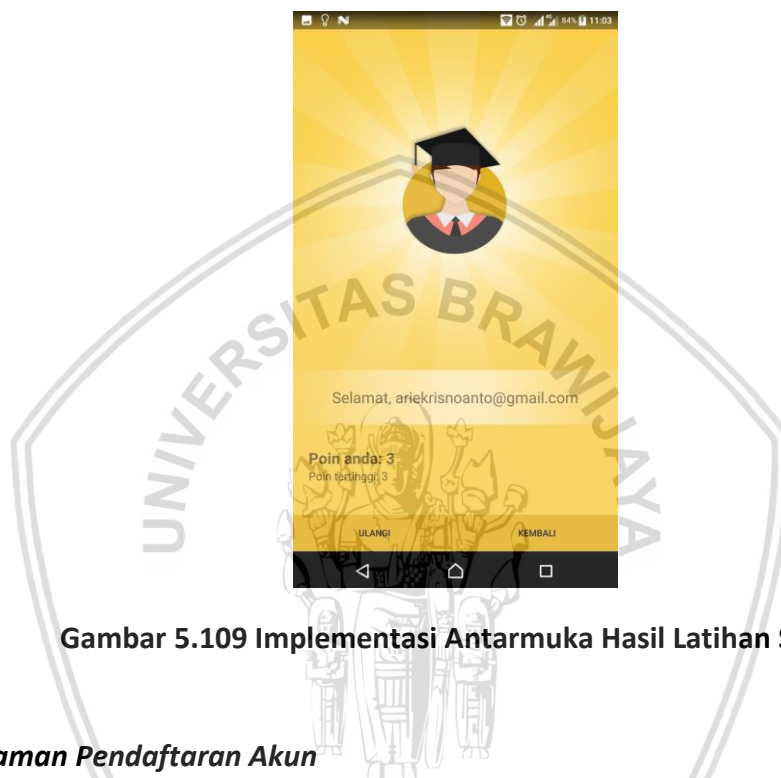
Gambar 5.107 Implementasi Antarmuka Latihan Soal 3



Gambar 5.108 Implementasi Antarmuka Latihan Soal 4

(q) Halaman Hasil Latihan Soal

Halaman hasil latihan soal ini memunculkan informasi berupa email pengguna saat itu, perolehan poin saat itu dan perolehan poin tertinggi yang pernah didapat oleh pengguna saat itu. Pada bagian bawah laman juga terdapat dua tombol yaitu tombol ulang dan tombol kembali, tombol ulang digunakan untuk melakukan latihan soal ulang dengan soal yang sama, dan tombol kembali digunakan untuk menuju ke halaman menu utama pengguna. Halaman hasil latihan soal ditunjukkan pada Gambar 5.109.



Gambar 5.109 Implementasi Antarmuka Hasil Latihan Soal

(r) Halaman Pendaftaran Akun

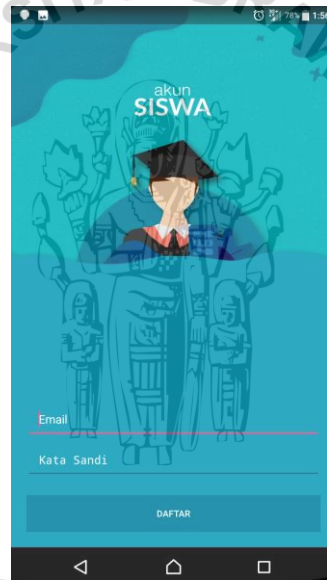
Halaman pendaftaran akun ini menampilkan tombol untuk menuju ke halaman pendaftaran siswa, laman ini diperlukan untuk mengurangi kesalahpahaman kepada pendaftar. Halaman pendaftaran akun ditunjukkan pada Gambar 5.110.

(s) Halaman Pendaftaran

Halaman pendaftaran ini berada pada sisi pengajar, terdapat *form* untuk memasukkan data berupa email dan kata sandi. Halaman pendaftaran dapat dilihat pada Gambar 5.111.



Gambar 5.110 Implementasi Antarmuka Pendaftaran Akun



Gambar 5.111 Implementasi Antarmuka Pendaftaran

BAB 6 PENGUJIAN

Pada bab ini akan dibahas tentang pengujian sistem E-Learning SMANTIGDA menggunakan pendekatan *User Centered Design*. Pengujian yang dilakukan terdapat dua tahap yaitu pengujian fungsional dan pengujian non-fungsional. Pengujian fungsional adalah pengujian yang dilakukan untuk menguji kebutuhan fungsional pada sistem, sedangkan pengujian non-fungsional dilakukan untuk menguji kebutuhan non-fungsional pada sistem. Proses pengujian fungsional yang dilakukan terhadap penelitian ini terdapat dua jenis pengujian yaitu pengujian *white box* dengan strategi pengujian unit dan pengujian integrasi, kemudian pengujian *black box* dengan strategi pengujian validasi. Kemudian untuk pengujian kebutuhan non-fungsional akan menggunakan pengujian *usability* dan *compability*.

6.1 Pengujian Fungsional

Pengujian fungsional digunakan untuk menguji seluruh kebutuhan fungsional yang telah dikembangkan dalam sistem. Pengujian ini terbagi menjadi pengujian unit, pengujian integrasi dan pengujian validasi. Pada tahap ini pengujian unit dan pengujian integrasi hanya dilakukan pada *method* dari fungsi utama sistem saja.

6.1.1 Pengujian Unit

Aplikasi yang telah dikembangkan dengan paradigma *object-oriented programming* ini menerapkan pengujian unit untuk suatu *method* dari suatu *class*. Pengujian unit aplikasi E-learning Smantigda ini menggunakan *white-box testing* dengan teknik *basis path testing*. Teknik *basis path testing* ini memodelkan algoritme pada suatu *flow graph* dan menentukan jumlah kompleksitas siklomatis, menentukan sebuah basis set dari jalur independen dan memberi kasus uji pada setiap basis set yang telah ditentukan sebelumnya.

6.1.1.1 Pengujian Method *getPDF()*

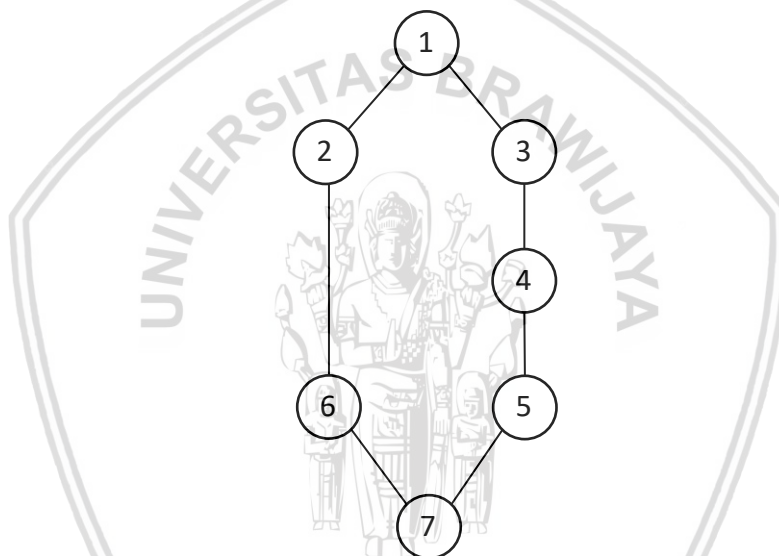
Method *getPDF* merupakan suatu unit *method* dari *class* *UnggahTugas* yang berfungsi untuk melakukan pengambilan berkas data pada memori penyimpanan. *Method* *getPDF* ditunjukkan oleh Gambar 6.1 berikut:



5	<pre> Toast.LENGTH_SHORT).show(); Intent intent = new Intent(UnggahTugas.this, TugasSiswa.class); startActivity(intent); } Intent intent = new Intent(); intent.setType("application/pdf"); intent.setAction(Intent.ACTION_GET_CONTENT); startActivityForResult (Intent.createChooser(intent, "Select Document"), PICK_PDF_CODE); } </pre>
6	
7	

Gambar 6.1 Kode Program method getPDF()

Berdasarkan proses pengujian yang telah dilakukan seperti pada gambar 6.1, maka *flowgraph* yang dihasilkan dari *method* getPDF sebagai berikut:



Gambar 6.2 Flowgraph method getPDF()

Tahapan selanjutnya yang dilakukan adalah menghitung *cyclomatic complexity*, cara melakukan perhitungan *cyclomatic complexity* menggunakan persamaan $V(G) = E - N + 2$ dimana $V(G)$ merupakan jumlah kompleksitas siklomatis, E (*edge*) merupakan garis penghubung antar node dan N itu sendiri merupakan jumlah simpul yang ada (*node*).

$$\begin{aligned}
 V(G) &= E - N + 2 \\
 &= 7 - 7 + 2 \\
 &= 0 + 2 \\
 &= 2
 \end{aligned}$$

Berdasarkan hasil dari perhitungan *cyclomatic complexity* didapatkan dua buah basis set jalur *independent*, yaitu:

Jalur 1 : 1 – 2 – 6 – 7

Jalur 2 : 1 – 3 – 4 – 5 – 7

Berdasarkan jumlah jalur yang didapatkan dari perhitungan yang telah dilakukan, dari jumlah jalur yang dihasilkan dibuat kasus ujinya sesuai jumlah jalur dan kondisi jalannya kode program pada method yang diuji. maka selanjutnya akan dilakukan pengujian terhadap tiap jalur tersebut yang ditunjukkan oleh Tabel 6.1 dibawah.

Tabel 6.1 Kasus Uji Pengujian Unit *method* getPDF()

Jalur	Kasus Uji	Hasil yang Diharapkan	Hasil yang Didapatkan
1	Ketika versi SDK sama atau telah mencukupi sesuai ketentuan dengan Android versi <i>Lollipop</i>	Sistem akan mengarah pada halaman pengaturan untuk membuka akses perizinan pada aplikasi untuk membaca memori penyimpanan dan melakukan intent aplikasi <i>file manager</i>	Sistem mengarah pada halaman pengaturan untuk membuka akses perizinan pada aplikasi untuk membaca memori penyimpanan dan melakukan intent aplikasi <i>file manager</i> .
2	Ketika versi SDK tidak mencukupi atau dibawah Android versi <i>Lollipop</i>	Sistem akan memunculkan pesan bahwa aplikasi tidak kompatibel dengan <i>device</i> yang sedang digunakan	Sistem memunculkan pesan bahwa aplikasi tidak kompatibel dengan <i>device</i> yang sedang digunakan.

6.1.1.2 Pengujian Method *uploadFile()*

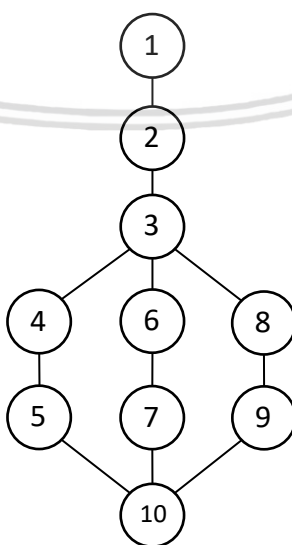
Method `uploadFile` adalah salah satu unit *method* dari class `UnggahTugas` yang berfungsi untuk memproses berkas yang telah dipilih untuk selanjutnya diunggah ke *database*. *Method* `uploadFile` ditunjukkan oleh Gambar 6.2 berikut:

1	{	<code>private void uploadFile(Uri data) {</code>
2		<code> progressBar.setVisibility(View.VISIBLE);</code>
3		<code> StorageReference sRef = mStorageReference.child</code>
4		<code> (Constant.STORAGE_PATH_UPLOADS + System</code>
5		<code> .currentTimeMillis() + ".pdf");</code>
		<code> sRef.putFile(data)</code>
		<code> .addOnSuccessListener(new OnSuccessListener</code>
		<code> <UploadTask.TaskSnapshot>() {</code>
		<code> @SuppressWarnings("VisibleForTests")</code>
		<code> @Override</code>
		<code> public void onSuccess(UploadTask</code>
		<code> .TaskSnapshot taskSnapshot) {</code>
		<code> progressBar.setVisibility(View.GONE);</code>
		<code> statusUnggah.setText("Tugas berhasil</code>

	<pre> diunggah"); ModelUpload upload = new ModelUpload (namaBerkas.getText().toString(), taskSnapshot.getDownloadUrl() .toString()); mDatabaseReference.child (mDatabaseReference.push() .getKey()).setValue(upload); } }) .addOnFailureListener(new OnFailureListener() { @Override public void onFailure(@NonNull Exception exception) { Toast.makeText(UnggahTugas.this, "Gagal Unggah!", Toast.LENGTH_LONG).show(); } }) .addOnProgressListener(new OnProgressListener <UploadTask.TaskSnapshot>() { @SuppressWarnings ("VisibleForTests") @Override public void onProgress (UploadTask.TaskSnapshot taskSnapshot) { double progress = (100.0 * taskSnapshot.getBytesTransferred()) / taskSnapshot.getTotalByteCount(); statusUnggah.setText((int) progress + "% Uploading..."); } }); } } </pre>
6	
7	
8	
9	
10	

Gambar 6.3 Kode Program method uploadFile()

Berdasarkan proses pengujian yang telah dilakukan seperti pada Gambar 6.3, maka *flowgraph* yang dihasilkan dari *method* uploadFile sebagai berikut



Gambar 6.4 Flowgraph method uploadFile()

Tahapan selanjutnya yang dilakukan adalah menghitung *cyclomatic complexity*, cara melakukan perhitungan *cyclomatic complexity* menggunakan persamaan $V(G) = E - N + 2$ dimana $V(G)$ merupakan jumlah kompleksitas siklomatis, E (*edge*) merupakan garis penghubung antar node dan N itu sendiri merupakan jumlah simpul yang ada (*node*).

$$\begin{aligned} V(G) &= E - N + 2 \\ &= 11 - 10 + 2 \\ &= 1 + 2 \\ &= 3 \end{aligned}$$

Berdasarkan hasil dari perhitungan *cyclomatic complexity* didapatkan tiga buah basis set jalur *independent*, yaitu:

Jalur 1 : 1 – 2 – 3 – 4 – 5 – 10

Jalur 2 : 1 – 2 – 3 – 6 – 7 – 10

Jalur 3 : 1 – 2 – 3 – 8 – 9 – 10

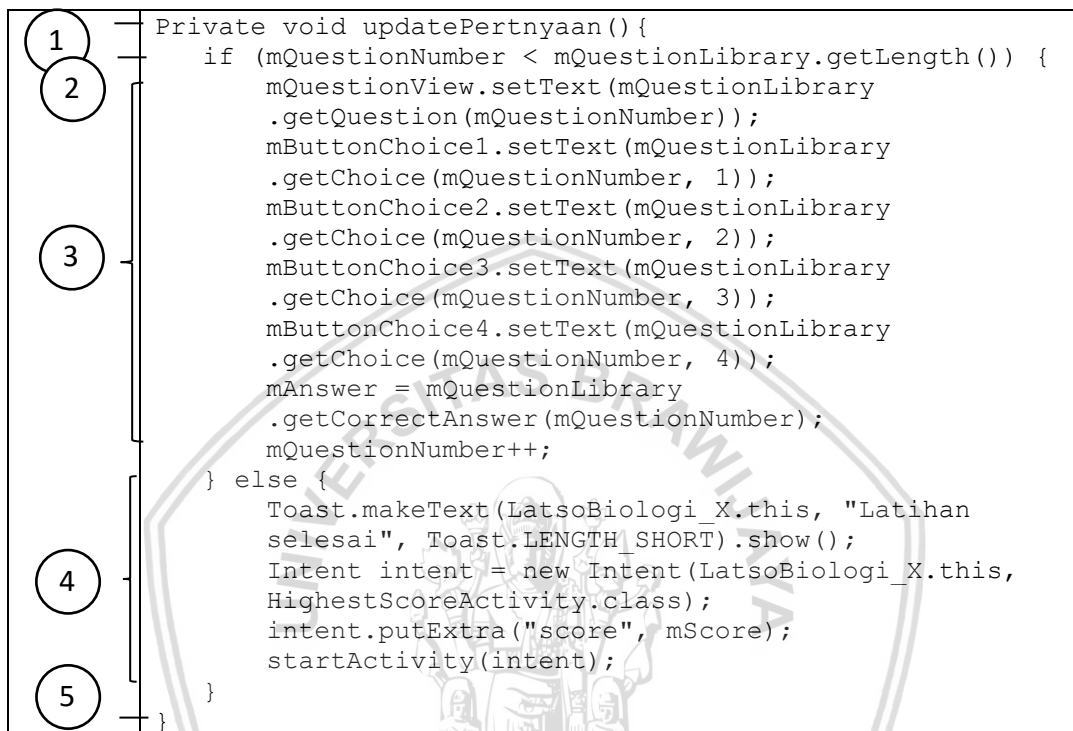
Berdasarkan jumlah jalur yang didapatkan dari perhitungan yang telah dilakukan, dari jumlah jalur yang dihasilkan dibuat kasus ujinya sesuai jumlah jalur dan kondisi jalannya kode program pada method yang diuji, maka selanjutnya akan dilakukan pengujian terhadap tiap jalur tersebut yang ditunjukkan oleh Tabel 6.2 dibawah:

Tabel 6.2 Kasus Uji Pengujian Unit *method* uploadFile()

Jalur	Kasus Uji	Hasil yang Diharapkan	Hasil yang Didapatkan
1	Ketika sistem dapat mengenali berkas dan mulai melakukan aktivitas unggah berkas tugas	Sistem akan mengirim berkas tugas ke <i>database</i> dan memunculkan pesan berhasil	Sistem mengirim berkas tugas ke database dan memunculkan pesan "Tugas Berhasil Diunggah"
2	Ketika sistem gagal melakukan aktivitas unggah berkas tugas	Sistem akan menampilkan pesan kegagalan	Sistem menampilkan pesan "Gagal unggah!"
3	Ketika sistem sedang dalam melakukan aktivitas unggah berkas tugas	Sistem akan menampilkan proses dan prosentase unggah berkas ke <i>database</i>	Sistem menampilkan proses dan prosentase unggah berkas ke <i>database</i>

6.1.1.3 Pengujian Method *updatePertanyaan()*

Method *updatePertanyaan* merupakan sebuah unit *method* dari class *LatsoBiologi_X*, dan juga ada di beberapa class latihan soal mata pelajaran lainnya. Method ini berfungsi untuk menampilkan beberapa pertanyaan latihan soal yang telah disediakan pada sistem. Method *updatePertanyaan* ditunjukkan oleh Gambar 6.5 berikut.

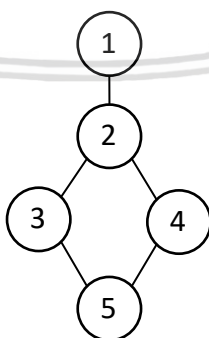


```

1  Private void updatePertanyaan(){
2      if (mQuestionNumber < mQuestionLibrary.getLength()) {
3          mQuestionView.setText(mQuestionLibrary
4              .getQuestion(mQuestionNumber));
5          mButtonChoice1.setText(mQuestionLibrary
              .getChoice(mQuestionNumber, 1));
              mButtonChoice2.setText(mQuestionLibrary
                  .getChoice(mQuestionNumber, 2));
                  mButtonChoice3.setText(mQuestionLibrary
                      .getChoice(mQuestionNumber, 3));
                      mButtonChoice4.setText(mQuestionLibrary
                          .getChoice(mQuestionNumber, 4));
                          mAnswer = mQuestionLibrary
                              .getCorrectAnswer(mQuestionNumber);
                              mQuestionNumber++;
                              } else {
                                  Toast.makeText(LatsoBiologi_X.this, "Latihan
                                      selesai", Toast.LENGTH_SHORT).show();
                                      Intent intent = new Intent(LatsoBiologi_X.this,
                                          HighestScoreActivity.class);
                                          intent.putExtra("score", mScore);
                                          startActivity(intent);
                                  }
          }
    
```

Gambar 6.5 Kode Program method *updatePertanyaan*

Berdasarkan proses pengujian yang telah dilakukan seperti pada Gambar 6.5, maka *flowgraph* yang dihasilkan dari *method* *updatePertanyaan* sebagai berikut.



Gambar 6.6 Flowgraph method *updatePertanyaan()*

Tahapan selanjutnya yang dilakukan adalah menghitung *cyclomatic complexity*, cara melakukan perhitungan *cyclomatic complexity* menggunakan persamaan $V(G) = E - N + 2$ dimana $V(G)$ merupakan jumlah kompleksitas

siklomatis, E (*edge*) merupakan garis penghubung antar node dan N itu sendiri merupakan jumlah simpul yang ada (*node*).

$$\begin{aligned} V(G) &= E - N + 2 \\ &= 5 - 5 + 2 \\ &= 2 \end{aligned}$$

Berdasarkan hasil dari perhitungan *cyclomatic complexity* didapatkan tiga buah basis set jalur *independent*, yaitu:

Jalur 1 : 1 – 2 – 3 – 5

Jalur 2 : 1 – 2 – 4 – 5

Berdasarkan jumlah jalur yang didapatkan dari perhitungan yang telah dilakukan maka, dari jumlah jalur yang dihasilkan dibuat kasus ujinya sesuai jumlah jalur dan kondisi jalannya kode program pada method yang diuji, selanjutnya akan dilakukan pengujian terhadap tiap jalur tersebut yang ditunjukkan oleh tabel 6.3 dibawah:

Tabel 6.3 Kasus Uji Pengujian Unit *method* updatePertanyaan()

Jalur	Kasus Uji	Hasil yang Diharapkan	Hasil yang Didapatkan
1	Ketika mQuestionNumber memiliki nilai <i>value</i> lebih kecil dari mQuestionLibrary	Sistem akan menampilkan kelanjutan soal, dan seterusnya	Sistem menampilkan soal soal selanjutnya
2	Ketika mQuestionNumber memiliki nilai <i>value</i> lebih sama atau lebih besar dari mQuestionLibrary	Sistem akan menampilkan pesan latihan selesai dan menampilkan skornya	Sistem menampilkan pesan latihan selesai dan menampilkan skornya

6.1.1.4 Pengujian Method *hitungSkor()*

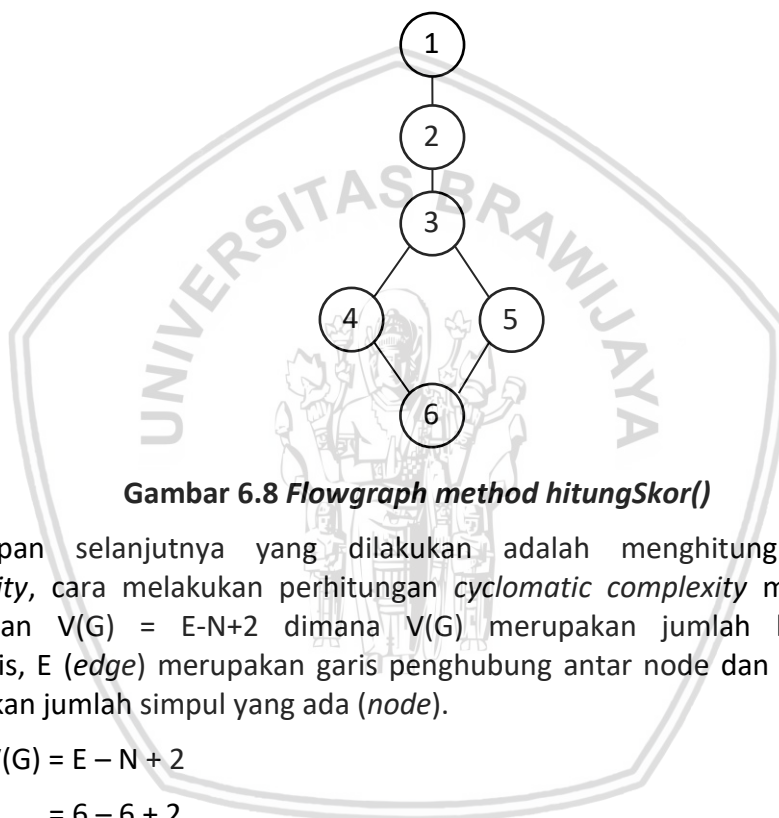
Method *hitungSkor* merupakan sebuah unit *method* dari *class* *LatsoBiologi_X*, dan juga ada di beberapa *class* latihan soal mata pelajaran lainnya. *Method* ini berfungsi untuk melakukan perhitungan skor saat pengguna menjawab pertanyaan dari latihan soal yang diberikan. *Method* *hitungSkor* ditunjukkan oleh Gambar 6.7.

1	private void hitungSkor(View view) {
2	Button answer = (Button) view;
3	if (answer.getText().equals(mAnswer)) {
4	mScore = mScore + 1;
	Toast.makeText(LatsoBiologi X.this, "Benar",

<div style="display: flex; flex-direction: column; align-items: center;"> <div style="border: 1px solid black; border-radius: 50%; width: 40px; height: 40px; display: flex; align-items: center; justify-content: center; margin-bottom: 10px;">5</div> <div style="border: 1px solid black; border-radius: 50%; width: 40px; height: 40px; display: flex; align-items: center; justify-content: center;">6</div> </div>	<pre> Toast.LENGTH_SHORT).show(); } else { Toast.makeText(LatsoBiologi_X.this, "Salah", Toast.LENGTH_SHORT).show(); updateSkor(mScore); } } </pre>
---	--

Gambar 6.7 Kode Program *method* hitungSkor()

Berdasarkan proses pengujian yang telah dilakukan seperti pada Gambar 6.7, maka *flowgraph* yang dihasilkan dari *method* hitungSkor sebagai berikut.



Gambar 6.8 Flowgraph *method* hitungSkor()

Tahapan selanjutnya yang dilakukan adalah menghitung *cyclomatic complexity*, cara melakukan perhitungan *cyclomatic complexity* menggunakan persamaan $V(G) = E - N + 2$ dimana $V(G)$ merupakan jumlah kompleksitas siklomatis, E (*edge*) merupakan garis penghubung antar node dan N itu sendiri merupakan jumlah simpul yang ada (*node*).

$$\begin{aligned}
 V(G) &= E - N + 2 \\
 &= 6 - 6 + 2 \\
 &= 2
 \end{aligned}$$

Berdasarkan hasil dari perhitungan *cyclomatic complexity* didapatkan tiga buah basis set jalur *independent*, yaitu:

Jalur 1 : 1 – 2 – 3 – 4 – 6

Jalur 2 : 1 – 2 – 3 – 5 – 6

Berdasarkan jumlah jalur yang didapatkan dari perhitungan yang telah dilakukan maka, dari jumlah jalur yang dihasilkan dibuat kasus ujinya sesuai jumlah jalur dan kondisi jalannya kode program pada *method* yang diuji, selanjutnya akan dilakukan pengujian terhadap tiap jalur tersebut yang ditunjukkan oleh Tabel 6.2:

Tabel 6.4 Kasus Uji Pengujian Unit *method* hitungSkor()

Jalur	Kasus Uji	Hasil yang Diharapkan	Hasil yang Didapatkan
1	Ketika jawaban telah diterima sistem dan mendapatkan hasil yang benar	Sistem akan menampilkan penambahan skor	Sistem menampilkan penambahan skor
2	Ketika jawaban diterima dan hasilnya salah	Sistem akan menampilkan soal selanjutnya	Sistem menampilkan soal selanjutnya

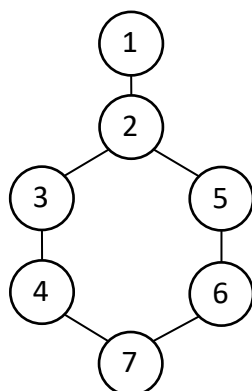
6.1.1.5 Pengujian Method cetakData()

Method cetakData merupakan sebuah unit *method* dari class HighestScoreActivity. *Method* ini berfungsi untuk melakukan cetak data perolehan skor dari latihan soal. *Method* cetakData ditunjukkan oleh Gambar 6.9 berikut:

1	private void cetakData(){
	Intent intent = getIntent();
	int score = intent.getIntExtra("score", 0);
	TextView txtScore = (TextView)
	findViewById(R.id.textScore);
	TextView txtHighScore = (TextView)
	findViewById(R.id.textHighScore);
2	txtScore.setText("Poin anda: " + score);
	SharedPreferences mypref =
	getPreferences(MODE_PRIVATE);
	int highscore = mypref.getInt("highscore",0);
3	if (highscore >= score)
4	txtHighScore.setText("Poin tertinggi:
5	txtHighScore.setText("Rekor poin tertinggi: "+score);
6	SharedPreferences.Editor editor = mypref.edit();
	editor.putInt("highscore", score);
7	editor.commit();
	}

Gambar 6.9 Kode Program *method* cetakData

Berdasarkan proses pengujian yang telah dilakukan seperti pada gambar 6.9, maka *flowgraph* yang dihasilkan dari *method* cetakData sebagai berikut:



Gambar 6.10 Flowgraph method cetakData()

Tahapan selanjutnya yang dilakukan adalah menghitung *cyclomatic complexity*, cara melakukan perhitungan *cyclomatic complexity* menggunakan persamaan $V(G) = E - N + 2$ dimana $V(G)$ merupakan jumlah kompleksitas siklomatis, E (*edge*) merupakan garis penghubung antar node dan N itu sendiri merupakan jumlah simpul yang ada (*node*).

$$\begin{aligned}
 V(G) &= E - N + 2 \\
 &= 7 - 7 + 2 \\
 &= 0 + 2 \\
 &= 2
 \end{aligned}$$

Berdasarkan hasil dari perhitungan *cyclomatic complexity* didapatkan dua buah basis set jalur *independent*, yaitu:

Jalur 1 : 1 – 2 – 3 – 4 – 7

Jalur 2 : 1 – 2 – 5 – 6 – 7

Berdasarkan jumlah jalur yang didapatkan dari perhitungan yang telah dilakukan maka dari jumlah jalur yang dihasilkan dibuat kasus ujinya sesuai jumlah jalur dan kondisi jalannya kode program pada method yang diuji, selanjutnya akan dilakukan pengujian terhadap tiap jalur tersebut yang ditunjukkan oleh Tabel 6.5 dibawah:

Tabel 6.5 Kasus Uji Pengujian Unit *method* cetakData()

Jalur	Kasus Uji	Hasil yang Diharapkan	Hasil yang Didapatkan
1	Ketika nilai <i>highscore</i> lebih atau sama dengan nilai <i>score</i> yang didapat	Sistem akan menampilkan poin tertinggi yang didapat	Sistem menampilkan poin tertinggi yang didapat
2	Ketika nilai yang didapat lebih dari	Sistem akan menampilkan rekor	Sistem menampilkan rekor poin tertinggi yang

	<i>score</i> dan <i>highscore</i>	poin tertinggi yang pernah didapat	pernah didapat
--	-----------------------------------	------------------------------------	----------------

6.1.2 Pengujian Validasi

Pegujian ini digunakan untuk mengetahui apakah sistem yang dibangun sudah sesuai dengan daftar kebutuhan. Pada pengujian ini terdapat beberapa kasus uji berdasarkan banyaknya kebutuhan fungsional yang terdapat pada sistem. Pengujian validasi menggunakan metode pengujian *Black Box*. Berikut kasus uji dan hasil pengujian validasi pada aplikasi *E-learning* SMANTIGDA.

6.1.2.1 Kasus Uji Validasi Pengguna Siswa

(a) Masuk ke halaman Login

Tabel 6.6 Kasus Uji Masuk ke Laman Login

Nama Kasus Uji	Masuk ke halaman Login
Tujuan Pengujian	Untuk membuktikan bahwa pengguna dapat masuk ke halaman login
Prosedur Pengujian	1. Memilih tombol siswa
Hasil yang Diharapkan	Pengguna dapat masuk ke halaman login siswa

(b) Login pengguna sebagai siswa

Tabel 6.7 Kasus Uji Masuk Sebagai Siswa

Nama Kasus Uji	Login pengguna sebagai siswa
Tujuan Pengujian	Untuk membuktikan bahwa pengguna siswa dapat masuk ke sistem
Prosedur Pengujian	1. Memasukkan <i>email</i> pengguna 2. Memasukkan kata sandi pengguna 3. Menekan tombol masuk
Hasil yang Diharapkan	Pengguna siswa dapat masuk dan sistem menampilkan halaman utama siswa

(c) Lihat Jadwal Pelajaran

Tabel 6.8 Kasus Uji Lihat Jadwal Pelajaran

Nama Kasus Uji	Lihat jadwal pelajaran
Tujuan Pengujian	Untuk membuktikan bahwa pengguna dapat melihat jadwal pelajaran
Prosedur Pengujian	1. Menekan tombol menu jadwal pelajaran

Hasil yang Diharapkan	Sistem menampilkan halaman jadwal pelajaran.
------------------------------	--

(d) Lihat info perubahan jadwal

Tabel 6.9 Kasus Uji Lihat Info Perubahan Jadwal

Nama Kasus Uji	Lihat info perubahan jadwal
Tujuan Pengujian	Untuk membuktikan bahwa pengguna dapat melihat info perubahan jadwal
Prosedur Pengujian	1. Menekan tombol menu jadwal pelajaran
Hasil yang Diharapkan	Sistem menampilkan daftar info perubahan jadwal

(e) Lihat Materi Pelajaran

Tabel 6.10 Kasus Uji Lihat Materi Pelajaran

Nama Kasus Uji	Lihat materi pelajaran
Tujuan Pengujian	Untuk membuktikan bahwa pengguna dapat melihat materi pelajaran
Prosedur Pengujian	1. Menekan tombol menu materi pelajaran 2. Menekan tombol pilihah kelas 3. Menekan tombol pilihan keminatan 4. Memilih mata pelajaran
Hasil yang Diharapkan	Sistem menampilkan materi pelajaran sesuai pilihan pengguna

(f) Lihat Daftar Tugas

Tabel 6.11 Kasus Uji Lihat Daftar Tuags

Nama Kasus Uji	Lihat daftar tugas
Tujuan Pengujian	Untuk membuktikan bahwa pengguna dapat melihat daftar tugas
Prosedur Pengujian	1. Menekan tombol Tugas pada menu utama
Hasil yang Diharapkan	Sistem menampilkan halaman tugas dan menampilkan daftar tugas

(g) Melakukan Unggah Tugas

Tabel 6.12 Kasus Uji Lihat Melakukan Unggah Tugas

Nama Kasus Uji	Melakukan unggah tugas
Tujuan Pengujian	Untuk membuktikan bahwa pengguna dapat melaukan unggah tugas

Prosedur Pengujian	<ol style="list-style-type: none"> 1. Menuliskan teks berkas tugas sesuai format 2. Menekan tombol pilih berkas 3. Memilih berkas yang akan diunggah melalui <i>file manager</i>
Hasil yang Diharapkan	Sistem mengunggah berkas tugas yang telah dipilih oleh pengguna

(h) Lihat Timeline Akademik

Tabel 6.13 Kasus Uji Lihat *Timeline* Akademik

Nama Kasus Uji	Lihat <i>timeline</i> akademik
Tujuan Pengujian	Untuk membuktikan bahwa pengguna dapat melihat <i>timeline</i> akademik
Prosedur Pengujian	<ol style="list-style-type: none"> 1. Menekan tombol <i>Timeline Akademik</i> pada menu utama
Hasil yang Diharapkan	Sistem menampilkan halaman <i>timeline</i> akademik.

(i) Lihat Daftar Aspirasi

Tabel 6.14 Kasus Uji Lihat Daftar Aspirasi

Nama Kasus Uji	Lihat daftar aspirasi
Tujuan Pengujian	Untuk membuktikan bahwa pengguna dapat melihat daftar aspirasi
Prosedur Pengujian	<ol style="list-style-type: none"> 1. Menekan tombol Aspirasi pada menu utama
Hasil yang Diharapkan	Sistem menampilkan halaman aspirasi

(j) Kirim Aspirasi

Tabel 6.15 Kasus Uji Mengirimkan Pesan Aspirasi

Nama Kasus Uji	Kirim aspirasi
Tujuan Pengujian	Untuk membuktikan bahwa pengguna dapat melakukan pengiriman pesan aspirasi
Prosedur Pengujian	<ol style="list-style-type: none"> 1. Menekan tombol Aspirasi pada menu utama 2. Memilih sasaran aspirasi 3. Menuliskan pesan aspirasi 4. Menekan tombol kirim
Hasil yang Diharapkan	Sistem mengirimkan dan menyimpan pesan aspirasi, dan menampilkan pesan aspirasi pada daftar aspirasi

(k) Lihat Profil Akun

Tabel 6.16 Kasus Uji Lihat Profil Akun Pengguna

Nama Kasus Uji	Lihat profil akun
Tujuan Pengujian	Untuk membuktikan bahwa pengguna dapat melihat profil akun
Prosedur Pengujian	1. Menekan tombol akun pada menu utama
Hasil yang Diharapkan	Sistem menampilkan halaman akun.

(l) Ubah Profil Akun

Tabel 6.17 Kasus Uji Melakukan Ubah Profil Akun

Nama Kasus Uji	Ubah profil akun
Tujuan Pengujian	Untuk membuktikan bahwa pengguna dapat melakukan pembaharuan profil akun
Prosedur Pengujian	<ol style="list-style-type: none"> 1. Menekan tombol akun pada menu utama 2. Menekan tombol ubah profil 3. Menuliskan nama pengguna 4. Menuliskan tepat lahir pengguna 5. Memilih tanggal, bulan dan tahun lahir pengguna
Hasil yang Diharapkan	Sistem menyimpan data baru pengguna dan menampilkan halaman akun

(m) Melakukan Latihan Soal

Tabel 6.18 Kasus Uji Melakukan Latihan Soal

Nama Kasus Uji	Melakukan latihan soal
Tujuan Pengujian	Untuk membuktikan bahwa pengguna dapat melakukan latihan soal
Prosedur Pengujian	<ol style="list-style-type: none"> 1. Menekan tombol Latihan soal pada menu utama 2. Memilih tombol kelas 3. Memilih tombol keminatan kelas 4. Memilih tombol mata pelajaran
Hasil yang Diharapkan	Sistem menampilkan soal dan opsi jawaban pada halaman latihan soal.

(a) Keluar dari sistem**Tabel 6.19 Kasus Uji Keluar dari Sistem**

Nama Kasus Uji	Keluar dari sistem
Tujuan Pengujian	Untuk membuktikan bahwa pengguna dapat keluar dari sistem
Prosedur Pengujian	1. Menekan tombol akun pada menu utama 2. Menekan tombol keluar
Hasil yang Diharapkan	Sistem menampilkan halaman <i>login screen</i>

6.1.2.2 Kasus Uji Validasi Pengguna Pengajar**(a) Masuk ke halaman login****Tabel 6.20 Kasus Uji Masuk ke halaman Login**

Nama Kasus Uji	Masuk ke halaman Login
Tujuan Pengujian	Untuk membuktikan bahwa pengguna dapat masuk ke halaman login
Prosedur Pengujian	1. Memilih tombol pengajar
Hasil yang Diharapkan	Pengguna dapat masuk ke halaman login pengajar

(b) Login pengguna sebagai pengajar**Tabel 6.21 Kasus Uji Masuk dalam Sistem Sebagai Pengajar**

Nama Kasus Uji	Login pengguna sebagai pengajar
Tujuan Pengujian	Untuk membuktikan bahwa pengguna pengajar dapat masuk ke sistem
Prosedur Pengujian	1. Memasukkan <i>NIP</i> 2. Memasukkan email dan kata sandi pengguna 3. Menekan tombol masuk
Hasil yang Diharapkan	Pengguna pengajar dapat masuk dan sistem menampilkan halaman utama pengajar

(c) Lihat jadwal pelajaran**Tabel 6.22 Kasus Uji Lihat Jadwal Pelajaran**

Nama Kasus Uji	Lihat jadwal pelajaran
Tujuan Pengujian	Untuk membuktikan bahwa pengguna dapat melihat jadwal pelajaran
Prosedur Pengujian	1. Menekan tombol menu jadwal pelajaran

Hasil yang Diharapkan	Sistem menampilkan halaman jadwal pelajaran.
------------------------------	--

(d) Tambah info perubahan jadwal

Tabel 6.23 Kasus Uji Melakukan Tambah Info Perubahan Jadwal

Nama Kasus Uji	Tambah info perubahan jadwal
Tujuan Pengujian	Untuk membuktikan bahwa pengguna dapat melakukan tambah info perubahan jadwal
Prosedur Pengujian	<ol style="list-style-type: none"> 1. Menekan tombol menu jadwal pelajaran 2. Menekan tombol tambah info baru 3. Menuliskan informasi perubahan jadwal 4. Memilih tanggal, bulan dan tahun pembaharuan perubahan jadwal 5. Menekan tombol simpan
Hasil yang Diharapkan	Sistem menampilkan daftar info perubahan jadwal pada laman perubahan jadwal.

(e) Lihat materi pelajaran

Tabel 6.24 Kasus Uji Lihat Materi Pelajaran

Nama Kasus Uji	Lihat materi pelajaran
Tujuan Pengujian	Untuk membuktikan bahwa pengguna dapat melihat materi pelajaran
Prosedur Pengujian	<ol style="list-style-type: none"> 1. Menekan tombol menu materi pelajaran 2. Menekan tombol pilihah kelas 3. Menekan tombol pilihan keminatan 4. Memilih mata pelajaran
Hasil yang Diharapkan	Sistem menampilkan materi pelajaran sesuai pilihan pengguna

(f)Lihat daftar tugas

Tabel 6.25 Kasus Uji Lihat Daftar Tugas

Nama Kasus Uji	Lihat daftar tugas
Tujuan Pengujian	Untuk membuktikan bahwa pengguna dapat melihat daftar tugas
Prosedur Pengujian	<ol style="list-style-type: none"> 1. Menekan tombol Tugas pada menu utama
Hasil yang Diharapkan	Sistem menampilkan halaman tugas dan

	menampilkan daftar tugas
--	--------------------------

(g) Tambah tugas

Tabel 6.26 Kasus Uji Menambah Daftar Tugas

Nama Kasus Uji	Tambah tugas
Tujuan Pengujian	Untuk membuktikan bahwa pengguna dapat menambah daftar tugas
Prosedur Pengujian	<ol style="list-style-type: none"> 1. Menekan tombol Tugas pada menu utama 2. Memilih mata pelajaran 3. Menuliskan tema tugas 4. Menuliskan index tugas 5. Memilih kelas sasaran tugas 6. Memilih batas waktu pengumpulan tugas 7. Memilih batas tanggal pengumpulan tugas 8. Menuliskan deskripsi tugas 9. Menekan tombol tambah
Hasil yang Diharapkan	Sistem menambahkan daftar tugas baru pada laman tugas pengajar

(h) Lihat berkas tugas

Tabel 6.27 Kasus Uji Melihat Berkas Tugas

Nama Kasus Uji	Lihat berkas tugas
Tujuan Pengujian	Untuk membuktikan bahwa pengguna dapat melihat daftar berkas tugas
Prosedur Pengujian	<ol style="list-style-type: none"> 1. Menekan tombol Tugas pada menu utama 2. Menekan tombol lihat berkas
Hasil yang Diharapkan	Sistem menampilkan daftar berkas tugas dari siswa

(i) Unduh berkas tugas

Tabel 6.28 Kasus Uji Unduh Berkas Tugas

Nama Kasus Uji	Unduh
Tujuan Pengujian	Untuk membuktikan bahwa pengguna dapat melakukan unduh berkas tugas
Prosedur Pengujian	<ol style="list-style-type: none"> 1. Menekan tombol Tugas pada menu utama 2. Menekan tombol lihat berkas 3. Menekan salah satu daftar berkas
Hasil yang Diharapkan	Sistem mengunduh salah satu berkas tugas dari

	siswa yang telah dipilih
--	--------------------------

(j) Lihat timeline akademik

Tabel 6.29 Kasus Uji Melihat *Timeline* Akademik

Nama Kasus Uji	Lihat <i>timeline</i> akademik
Tujuan Pengujian	Untuk membuktikan bahwa pengguna dapat melihat <i>timeline</i> akademik
Prosedur Pengujian	1. Menekan tombol <i>Timeline</i> Akademik pada menu utama
Hasil yang Diharapkan	Sistem menampilkan halaman <i>timeline</i> akademik.

(k) Tambah kegiatan

Tabel 6.30 Kasus Uji Menambah Kegiatan Akademik

Nama Kasus Uji	Tambah kegiatan
Tujuan Pengujian	Untuk membuktikan bahwa pengguna dapat menambah kegiatan akademik
Prosedur Pengujian	1. Menekan tombol <i>timeline</i> akademik pada menu utama 2. Menekan tombol tambah kegiatan 3. Menulis judul kegiatan 4. Memilih tanggal, bulan dan tahun kegiatan 5. Menekan tombol tambah
Hasil yang Diharapkan	Sistem menyimpan kegiatan baru dan menampilkannya pada daftar <i>tilmeline</i> akademik

(l) Lihat daftar aspirasi

Tabel 6.31 Kasus Uji Melihat Daftar Aspirasi

Nama Kasus Uji	Lihat daftar aspirasi
Tujuan Pengujian	Untuk membuktikan bahwa pengguna dapat melihat daftar aspirasi
Prosedur Pengujian	1. Menekan tombol Aspirasi pada menu utama
Hasil yang Diharapkan	Sistem menampilkan halaman aspirasi

(m) Kirim Aspirasi**Tabel 6.32 Kasus Uji Melakukan Pengiriman Pesan Aspirasi**

Nama Kasus Uji	Kirim aspirasi
Tujuan Pengujian	Untuk membuktikan bahwa pengguna dapat melakukan pengiriman pesan aspirasi
Prosedur Pengujian	<ol style="list-style-type: none"> 1. Menekan tombol Aspirasi pada menu utama 2. Memilih sasaran aspirasi 3. Menuliskan pesan aspirasi 4. Menekan tombol kirim
Hasil yang Diharapkan	Sistem mengirimkan dan menyimpan pesan aspirasi, dan menampilkan pesan aspirasi pada daftar aspirasi

(n) Pendaftaran akun siswa**Tabel 6.33 Kasus Uji Melakukan Pendaftaran Akun Siswa**

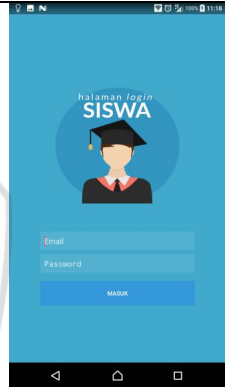
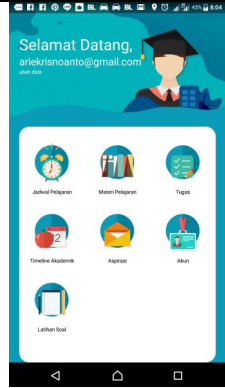
Nama Kasus Uji	Pendaftaran akun siswa
Tujuan Pengujian	Untuk membuktikan bahwa pengguna dapat melakukan pendaftaran akun siswa
Prosedur Pengujian	<ol style="list-style-type: none"> 1. Menekan tombol pendaftaran pada menu utama 2. Menekan tombol siswa pada laman pendaftaran 3. Memasukkan <i>email</i> pengguna siswa 4. Memasukkan kata sandi pengguna siswa
Hasil yang Diharapkan	Sistem membuat akun siswa baru

(o) Keluar dari sistem**Tabel 6.34 Kasus Uji Keluar dari Sistem**

Nama Kasus Uji	Keluar dari sistem
Tujuan Pengujian	Untuk membuktikan bahwa pengguna dapat keluar dari sistem
Prosedur Pengujian	<ol style="list-style-type: none"> 1. Menekan tombol keluar pada menu utama
Hasil yang Diharapkan	Sistem menampilkan halaman <i>login screen</i>

6.1.2.3 Hasil Pengujian Validasi Pengguna Siswa

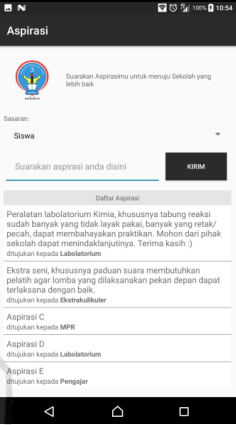
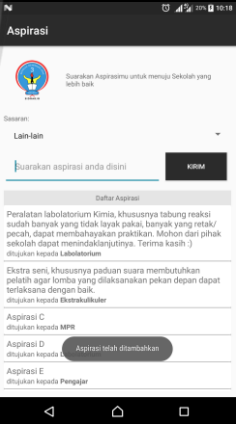
Tabel 6.35 Hasil Pengujian Validasi Pengguna Siswa

No	Nama Kasus Uji	Hasil yang Diharapkan	Hasil yang Didapatkan	Screenshot Validasi	Status Validasi
1	Masuk ke halaman <i>login</i>	Pengguna dapat masuk ke halaman login siswa	Pengguna dapat masuk ke halaman login siswa		<i>Valid</i>
2	<i>Login</i> pengguna sebagai siswa	Pengguna siswa dapat masuk dan sistem menampilkan halaman utama siswa	Pengguna siswa dapat masuk dan sistem menampilkan halaman utama siswa		<i>Valid</i>

3	Lihat jadwal pelajaran	Sistem menampilkan halaman jadwal pelajaran.	Sistem menampilkan halaman jadwal pelajaran.		Valid
4	Lihat info perubahan jadwal	Sistem menampilkan daftar info perubahan jadwal	Sistem menampilkan daftar info perubahan jadwal		Valid

5	Lihat materi pelajaran	Sistem menampilkan materi pelajaran sesuai pilihan pengguna	Sistem menampilkan materi pelajaran sesuai pilihan pengguna		Valid
6	Lihat daftar tugas	Sistem menampilkan halaman tugas dan menampilkan daftar tugas	Sistem menampilkan halaman tugas dan menampilkan daftar tugas		Valid

<p>pilih oleh pengguna</p>	<p>pengguna</p>
<p>stem menampilkan halaman <i>timeline</i> akademik.</p>	<p>Sistem menampilkan halaman <i>timeline</i> akademik.</p>

9	Lihat daftar aspirasi	Sistem menampilkan halaman aspirasi	Sistem menampilkan halaman aspirasi		Valid
10	Kirim aspirasi	Sistem mengirimkan dan menyimpan pesan aspirasi, dan menampilkan pesan aspirasi pada daftar aspirasi	Sistem mengirimkan dan menyimpan pesan aspirasi, dan menampilkan pesan aspirasi pada daftar aspirasi		Valid

11	Lihat profil akun	Sistem menampilkan halaman akun.	Sistem menampilkan halaman akun.		Valid
12	Ubah profil akun	Sistem menyimpan data baru pengguna dan menampilkan halaman akun	Sistem menyimpan data baru pengguna dan menampilkan halaman akun		Valid

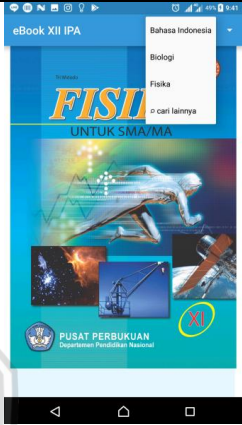
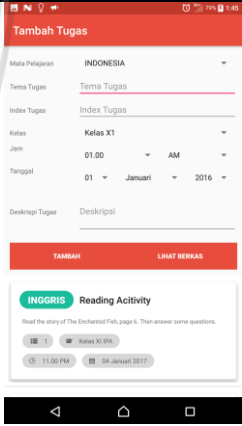
13	Melakukan latihan soal	Sistem menampilkan soal dan opsi jawaban pada halaman latihan soal.	Sistem menampilkan soal dan opsi jawaban pada halaman latihan soal.		Valid
14	Keluar dari sistem	Sistem menampilkan halaman <i>logon screen</i>	Sistem menampilkan halaman <i>logon screen</i>		Valid

6.1.2.4 Hasil Pengujian Validasi Pengguna Pengajar

Tabel 6.36 Hasil Pengujian Validasi Pengguna Pengajar

No	Nama Kasus Uji	Hasil yang Diharapkan	Hasil yang Didapatkan	Screenshot Validasi	Status Validasi
1	Masuk ke halaman <i>login</i>	Pengguna dapat masuk ke halaman <i>login</i> pengajar	Pengguna dapat masuk ke halaman <i>login</i> pengajar		<i>Valid</i>
2	<i>Login</i> pengguna sebagai pengajar	Pengguna pengajar dapat masuk dan sistem menampilkan halaman utama pengajar	Pengguna pengajar dapat masuk dan sistem menampilkan halaman utama pengajar		<i>Valid</i>

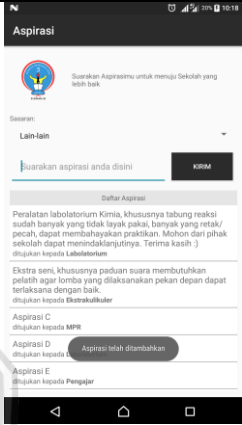

3	Lihat jadwal pelajaran	Sistem menampilkan halaman jadwal pelajaran.	Sistem menampilkan halaman jadwal pelajaran.		Valid
4	Tambah info perubahan jadwal	Sistem menampilkan daftar info perubahan jadwal pada laman perubahan jadwal.	Sistem menampilkan daftar info perubahan jadwal pada laman perubahan jadwal.		Valid

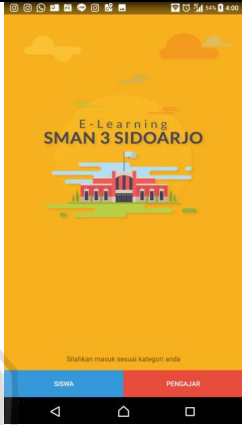
5	Lihat materi pelajaran	Sistem menampilkan materi pelajaran sesuai pilihan pengguna	Sistem menampilkan materi pelajaran sesuai pilihan pengguna		Valid
6	Lihat daftar tugas	Sistem menampilkan halaman tugas dan menampilkan daftar tugas	Sistem menampilkan halaman tugas dan menampilkan daftar tugas		Valid

7	Tambah tugas	Sistem menambahkan daftar tugas baru pada laman tugas pengajar	Sistem menambahkan daftar tugas baru pada laman tugas pengajar		Valid
8	Lihat berkas tugas	Sistem menampilkan daftar berkas tugas dari siswa	Sistem menampilkan daftar berkas tugas dari siswa		Valid

9	Unduh berkas tugas	Sistem mengunduh salah satu berkas tugas dari siswa yang telah dipilih	Sistem mengunduh salah satu berkas tugas dari siswa yang telah dipilih		Valid
10	Lihat <i>timeline</i> akademik	Sistem menampilkan halaman <i>timeline</i> akademik.	Sistem menampilkan halaman <i>timeline</i> akademik.		Valid

11	Tambah kegiatan	Sistem menyimpan kegiatan baru dan menampilkannya pada daftar <i>timeline</i> akademik	Sistem menyimpan kegiatan baru dan menampilkannya pada daftar <i>timeline</i> akademik		Valid
12	Lihat daftar aspirasi	Sistem menampilkan halaman aspirasi	Sistem menampilkan halaman aspirasi		Valid

13	Kirim aspirasi	Sistem mengirimkan dan menyimpan pesan aspirasi, dan menampilkan pesan aspirasi pada daftar aspirasi	Sistem mengirimkan dan menyimpan pesan aspirasi, dan menampilkan pesan aspirasi pada daftar aspirasi		Valid
14	Pendaftaran akun siswa	Sistem membuat akun siswa baru	Sistem membuat akun siswa baru		Valid

15	Keluar dari sistem	Sistem menampilkan halaman <i>login screen</i>	Sistem menampilkan halaman <i>login screen</i>		Valid
----	--------------------	--	--	---	-------

6.2 Pengujian Non-Fungsional

Pengujian non-fungsional juga dilakukan dalam penelitian ini, pengujian kebutuhan non-fungsional yang akan dilakukan adalah pengujian *usability* dan pengujian *compatibility*.

6.2.1 Pengujian *Usability*

Pengujian *usability* ini dilakukan sesuai dengan hasil analisis kebutuhan non-fungsional yang telah dijelaskan sebelumnya pada tabel 4.10. Pengujian *usability* pada aplikasi E-Learning SMANTIGDA ini untuk mengetahui tingkat kepuasan dan kemudahan dari pengguna aplikasi yang mengacu sesuai aspek-aspek *usability*. Pada pengujian *usability* terdapat dua metode pengujian yang dilakukan yaitu menggunakan *task scenario* dan *USE Quisissionnaire*. Pengujian *usability* yang pertama menggunakan metode *task scenario* yaitu dengan memberikan beberapa *task* atau tugas yang telah ditentukan sebelumnya dan diberikan kepada pengguna, kemudian untuk pengujian *usability* yang kedua menggunakan metode kuesioner *Usefulness, Satisfaction, Ease of Use (USE)* dengan memberikan 27 pertanyaan untuk seluruh responden. Pengujian ini melibatkan 18 orang responden yang terdiri dari 14 siswa dan 4 pengajar sesuai sasaran pengguna dari aplikasi yang dikembangkan.

6.2.1.1 Pengujian *Usability* Menggunakan *Task Scenario*

Metode *task scenario* yang dilakukan pada pengujian *usability* ini dimulai memberikan 13 *task* kepada pengguna siswa dan 14 *task* kepada pengguna pengajar yang harus dijalankan. Setelah pengguna selesai menjalankan aplikasi sesuai *task scenario* yang diberikan, kemudian dilakukan verifikasi apakah setiap *task* yang diberikan berhasil atau gagal dilakukan, jika didapati terjadi kegagalan pada aplikasi saat menjalankan *task* yang diberikan, cari tahu alasan apa yang menyebabkan kegagalan pengguna saat menjalankan *task scenario* yang diberikan tersebut. Daftar *task skenario* yang diberikan kepada pengguna untuk sisi siswa dijelaskan pada dalam Tabel 6.38 berikut.

Tabel 6.37 Daftar *Task* Pengujian *Task Scenario* (Siswa)

No	<i>Task</i>	<i>Task Deskripsi</i>
1	Masuk sebagai siswa	Siswa melakukan aktivitas login untuk masuk ke menu utama aplikasi
2	Melihat jadwal pelajaran, melihat info perubahan jadwal dan memilih kelas jadwal pelajaran	Siswa melihat jadwal pelajaran dan info perubahan jadwal pada laman dan memilih kelas untuk memilih jadwal yang berbeda dari kelas lain
3	Melihat materi pelajaran	Siswa melihat beberapa dokumen materi pelajaran pada aplikasi
4	Mencari materi pelajaran yang tidak	Siswa mencari materi pelajaran yang tidak terdapat pada aplikasi

	tersedia pada aplikasi	
5	Melihat daftar tugas	Siswa melihat tugas yang diberikan pada pengajar
6	Melakukan aktivitas unggah tugas	Siswa melakukan aktivitas unggah berkas tugas yang terdapat pada memori penyimpanan melalui aplikasi
7	Melihat <i>timeline</i> akademik	Siswa melihat daftar <i>timeline</i> akademik sekolah
8	Lihat daftar aspirasi	Siswa melihat daftar aspirasi yang dikirimkan oleh siswa dan pengajar pada aplikasi
9	Kirim aspirasi	Siswa melakukan aktivitas menyuarkan aspirasinya dengan aplikasi
10	Melihat profil pengguna	Siswa melihat detil profil pengguna siswa
11	Melakukan edit profil pengguna	Siswa melakukan perubahan detil profil pada aplikasi
12	Melakukan latihan soal	Siswa melakukan latihan soal sesuai pilihan pengguna
13	Melakukan keluar / <i>logout</i> dari aplikasi	Siswa melakukan aktivitas <i>logout</i> dari aplikasi

Dan berikut daftar *task skenario* yang diberikan kepada pengguna untuk sisi pengajar dijelaskan pada dalam Tabel 6.39.

Tabel 6.38 Daftar Task Pengujian Task Scenario (Pengajar)

No	Task Description	Task Scenario
1	Masuk sebagai pengajar	Pengajar melakukan aktivitas login pada <i>aplikasi</i>
2	Melihat jadwal pelajaran, melihat info perubahan jadwal dan memilih kelas jadwal pelajaran	Pengajar melihat beberapa jadwal pelajaran dari berbagai kelas dan info perubahan jadwal
3	Menambah info perubahan jadwal	Pengajar melakukan aktivitas tambah info perubahan jadwal
4	Melihat materi pelajaran	Pengajar melihat berbagai dokumen materi pelajaran yang telah disediakan oleh aplikasi
5	Mencari materi pelajaran yang tidak tersedia pada aplikasi	Pengajar mencari materi pelajaran yang tidak tersedia pada aplikasi
6	Melihat daftar tugas	Pengajar melihat daftar tugas yang pernah diberikan oleh pengajar itu sendiri kepada siswa
7	Melakukan aktivitas	Pengajar melakukan aktivitas tambah daftar

	tambah tugas	tugas pada aplikasi yang akan diberikan kepada siswa
8	Melakukan unduh berkas tugas	Pengajar melakukan aktivitas unduh beberapa berkas yang diberikan siswa kepada pengajar pada laman berkas tugas
9	Melihat <i>timeline</i> akademik	Pengajar melihat daftar kegiatan akademik pada aplikasi
10	Melakukan tambah kegiatan akademik	Pengajar melakukan aktivitas tambah kegiatan akademik aplikasi
11	Lihat daftar aspirasi	Pengajar melihat daftar aspirasi yang telah dikirimkan oleh siswa dan pengajar sebelumnya
12	Kirim aspirasi	Pengajar menyuarkan aspirasinya melalui fitur yang disediakan pada aplikasi
13	Melakukan aktivitas pendaftaran akun siswa	Pengajar mendaftarkan akun siswa baru melalui aplikasi
14	Melakukan keluar / <i>logout</i> dari aplikasi	Pengajar melakukan aktivitas <i>logout</i> dari sistem aplikasi

Pada tabel 6.38 terdapat hasil dari pengujian *usability* menggunakan *task scenario* dari pengguna siswa yang terdiri dari 14 responden, dan pada tabel 6.39 terdapat hasil dari pengujian *usability* menggunakan *task scenario* dari pengguna pengajar yang terdiri dari 4 responden. Sifat dari responden ini diambil secara random namun tetap bersifat representatif terhadap aplikasi yang sedang dikembangkan. Dari kedua pengguna terdapat masing-masing 13 *task* untuk siswa dan 14 *task* untuk pengajar yang harus dijalankan, dan *task* tersebut bersifat *goal based task scenario*.

Tabel 6.39 Hasil Pengujian Task Scenario (Siswa)

No.	Pengguna Siswa	Hasil Pengujian													Ket.
		T1	T2	T3	T4	T5	T6	T7	T8	T9	T10	T11	T12	T13	
1	Indradi T. P.	√	√	√	√	√	√	√	√	√	√	√	√	√	-
2	Aditya Setya	√	√	√	√	√	√	√	√	√	√	√	√	√	-
3	Mellania W.N.	√	√	√	√	√	X	√	√	√	√	√	√	√	C
4	Annisa Nur C.	√	√	√	√	√	√	√	√	√	√	√	√	√	-
5	Navyan N. A.	√	√	√	√	√	√	√	√	√	√	√	√	√	-
6	Andika P.	√	√	√	√	√	√	√	√	x	√	√	√	√	B
7	Annisa Azalia	√	√	√	√	√	X	√	√	√	√	√	√	√	C
8	I Gusti M. W.	√	√	√	√	√	√	√	√	√	√	√	√	√	-
9	Danial A.	√	√	√	√	√	√	√	√	√	√	√	√	√	-
10	Raisya G. M.	√	√	√	√	√	√	√	√	√	√	√	√	√	-
11	Dion Prakarsa	√	√	√	√	√	X	√	√	x	√	√	√	√	C, B
12	Dwike P.	√	√	√	√	√	X	√	√	√	√	√	√	√	C
13	Fahmi Dwi M.	√	√	√	√	√	√	√	√	√	√	√	√	√	-
14	Gunawan P.	√	√	√	√	√	√	√	√	√	√	√	√	√	-
Hasil		100%	100%	100%	100%	100%	71,4%	100%	100%	85,7%	100%	100%	100%	100%	

Tabel 6.40 Hasil Pengujian Task Scenario (Pengajar)

No.	Pengguna Pengajar	Hasil Pengujian														Ket.
		T1	T2	T3	T4	T5	T6	T7	T8	T9	T10	T11	T12	T13	T14	
1	Bp. Misbah	√	√	√	√	√	√	√	√	√	√	√	√	√	√	-
2	Bu Isa	√	√	√	√	√	√	x	√	√	√	√	√	√	√	B
3	Fraü Ariani	√	√	√	√	√	√	√	√	√	√	√	√	√	√	-
4	Bp. Ali T.	√	√	√	√	√	√	√	x	√	√	x	√	√	√	D, A
Hasil		100%	100%	100%	100%	100%	100%	75%	75%	100%	100%	75%	100%	100%	100%	

Pada masing-masing Tabel pengujian *usability* 6.40 dan 6.41 terdapat beberapa simbol dan abjad yang menunjukkan hasil dari pengujiannya. Untuk simbol (√) memberikan arti bahwa *task* yang dijalankan berhasil, untuk simbol (x) menandakan bahwa *task* yang dijalankan tidak berhasil atau gagal. Pada kolom keterangan terdapat beberapa keterangan yang mewakili keterangan dari alasan kegagalan yang diterima oleh pengguna, dan keterangan tersebut diantaranya:

- (A) Tidak ada koneksi internet / wifi
- (B) Tidak ada data yang dimasukkan
- (C) Tidak ada data yang cocok
- (D) Memori penyimpanan penuh

6.2.1.2 Pengujian Usability Menggunakan USE Quissionaire

USE Quissionaire digunakan untuk menguji secara langsung aplikasi yang telah dikembangkan oleh responden, setelah itu mengisi *form* kuesioner dengan beberapa pertanyaan yang diajukan. Tahap awal yang dipastikan sebelum responden mencoba aplikasinya yaitu memastikan *device* responden *compatible* dengan aplikasi yang dikembangkan, terdapat koneksi internet, terdapat berkas *.pdf yang tersimpan dalam *device* responden. Pengguna diberikan kebebasan untuk menggunakan aplikasi tanpa dibatasi oleh waktu, setelah selesai dan merasa cukup menggunakan aplikasinya, responden diarahkan untuk mengisi *form* kuesioner yang berisi 25 pertanyaan.

Hasil dari pengisian *USE Quissionaire* dapat dilihat pada Tabel 6.42.

Tabel 6.41 Hasil Kuesioner USE

USEFULLNESS		STS	TS	N	S	SS
1	Aplikasi ini efektif untuk membantu kegiatan akademik maupun pembelajaran	-	-	-	-	18
2	Aplikasi ini membantu untuk lebih produktif lagi dalam kegiatan akademik maupun pembelajaran	-	-	-	-	18
3	Aplikasi ini berguna untuk kegiatan akademik sekolah dan pembelajaran untuk siswa dan pengajar	-	-	-	1	17
4	Aplikasi ini membantu menyelesaikan pekerjaan saya lebih mudah	-	-	-	1	17
5	Aplikasi ini menghemat waktu untuk menyelesaikan pekerjaan saya	-	-	-	2	16
6	Aplikasi ini memenuhi kebutuhan saya dalam hal kegiatan dan pembelajaran untuk sekolah	-	-	-	1	17
7	Aplikasi ini menjalankan fungsinya sesuai dengan ekspektasi saya	-	-	1	1	16
EASE OF USE						
8	Aplikasi ini mudah dimengerti	-	-	-	2	16
9	Hanya sedikit tahapan yang dibutuhkan dalam menggunakan aplikasi ini	-	-	-	-	18
10	Dalam menggunakan aplikasi ini hanya membutuhkan sedikit usaha	-	-	1	3	14
11	Saya dapat menggunakan aplikasi ini	-	-	6	9	3


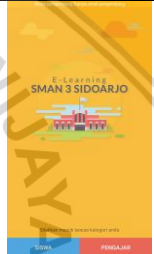

	tanpa petunjuk penggunaan					
12	Aplikasi ini berfungsi dengan konsisten ketika digunakan	-	-	4	12	2
13	Saya bisa mengatasi kesalahan ketika menggunakan aplikasi ini dengan cepat dan mudah	-	-	1	7	10
14	Saya berhasil menggunakan aplikasi ini setiap saat saya mengaksesnya	-	-	-	3	15
15	Pengguna aplikasi Elearning maupun pengguna yang jarang menggunakan aplikasi Elearning akan menyukai aplikasi ini	-	-	-	3	15
EASE OF LEARNING						
16	Saya mempelajari cara penggunaan aplikasi dengan cepat	-	-	-	-	18
17	Saya dengan mudah mengingat bagaimana cara menggunakan aplikasi ini	-	-	-	1	17
18	Saya mudah dalam mempelajari cara menggunakan aplikasi ini	-	-	-	-	18
19	Saya dengan cepat dan terampil dalam menggunakan aplikasi ini	-	-	-	1	17
SATISFACTION						
20	Saya puas dengan aplikasi ini	-	-	3	6	9
21	Saya akan merekomendasikan aplikasi ini kepada orang lain	-	-	-	4	14
22	Sangat menyenangkan dalam menggunakan aplikasi ini	-	-	-	2	16
23	Aplikasi ini bekerja sesuai dengan keinginan saya	-	-	2	5	11
24	Saya nyaman dalam menggunakan aplikasi ini	-	-	-	6	12
25	Saya harus memiliki aplikasi ini	-	-	-	2	16

Kuesioner USE ini mengacu pada jurnal *Measuring Usability with the USE Questionnaire* oleh Arnold M. Lund.

6.2.2 Pengujian *Compatibility*

Pengujian *compatibility* ini untuk melihat kompatibilitas dari aplikasi yang sedang dikembangkan. Pengujian ini untuk memenuhi kebutuhan non-fungsional *valid* atau tidaknya aplikasi berjalan pada beberapa versi sistem operasi Android.

Tabel 6.42 Hasil Uji Tes *Compatibility*

No	Perangkat	Screenshot Validasi	Status Validasi
1	Redmi note 3 - LCD : 5.5 inci, 720 x 1280 px - SDK : Android 5.0.2, API 21		<i>Valid</i>
2	Oppo A37 - LCD : 5.0 inci, 720 x 1280 px - SDK : Android 5.1.1, API 22		<i>Valid</i>
3	Sony Xperia XA Ultra Dual - LCD : 6.0 inci, 1080 x 1920 px - SDK : Android 7, API 24		<i>Valid</i>

6.3 Analisis Hasil Pengujian

Tahap ini menjelaskan tentang hasil analisis yang didapatkan dari beberapa pengujian yang dilakukan pada aplikasi E-learning SMANTIGDA pada perangkat Android. Analisis dilakukan terhadap semua hasil pengujian yang telah dilakukan yang meliputi analisis hasil pengujian unit, analisis hasil pengujian validasi, analisis hasil pengujian *usability* dan analisis hasil pengujian *compatibility*.

6.3.1 Analisis Hasil Pengujian Unit

Analisis hasil dari pengujian unit yang telah dilakukan dengan cara menggunakan *white-box testing* dengan *basis path testing* serta melihat apakah fungsi dari kebutuhan fungsional yang ada telah diimplementasikan dalam unit modul yang telah diuji memiliki kesesuaian dengan tahap perancangan yang telah dilakukan sebelumnya. Berdasarkan hal tersebut dan hasil pengujian yang

telah dilakukan, maka dapat disimpulkan bahwa unit modul dari aplikasi sudah memenuhi kebutuhan fungsional yang telah dirancang pada tahap perancangan.

6.3.2 Analisis Hasil Pengujian Validasi

Analisis hasil dari pengujian validasi dilakukan dengan membandingkan hasil uji dan daftar kebutuhan yang ada. Jika telah sesuai dengan skenario sistem, maka hasil tersebut *valid* karena telah berhasil melakukan implementasi kebutuhan fungsional. Apabila didapati hasil yang tidak *valid*, maka sistem tersebut belum memenuhi semua kebutuhan fungsionalnya. Berdasarkan pernyataan tersebut dapat disimpulkan bahwa hasil pengujian validasi E-learning SMANTIGDA telah memenuhi kebutuhan fungsional yang telah didefinisikan sebelumnya dan mayoritas telah menghasilkan hasil uji yang *valid*.

6.3.3 Analisis Hasil Pengujian Usability

Pada tahap ini terdapat penjelasan tentang analisis hasil pengujian *usability* dengan menggunakan *Task Scenario* dan *USE Quisinnnaire* yang dilakukan.

6.3.3.1 Analisis Hasil Pengujian Usability Menggunakan Task Scenario

Hasil dari pengujian *usability* menggunakan *task scenario* pada siswa yaitu mendapatkan hasil 100% pada seluruh *task* yang ada kecuali pada *task 6* dan *task 9* yang ditemukan lima orang siswa mengalami kegagalan sehingga presentase kesuksesan hanya mencapai 71,4 % untuk *task 6* dan 85,7% untuk *task 9*. Untuk *task 6* terdapat kegagalan saat menjalankan *task scenario* karena tidak adanya *file* atau berkas data yang cocok dengan format ber-ekstensi .pdf yang tersimpan pada memori penyimpanan *device* masing-masing pengguna, kemudian juga terjadi kegagalan pada *task 9* dikarenakan tidak melakukan *input* data dengan lengkap saat menjalankan *task scenario*.

Pada sisi pengajar hasil pengujian *usability* menggunakan *task scenario* juga tidak mendapatkan hasil presentase 100% pada keseluruhan *task*, melainkan terjadi kegagalan pada *task 7*, *task 8* dan *task 11* sehingga hanya mendapat 75% presentase keberhasilan. Penyebab terjadinya kegagalan pada *task 7* yaitu tidak lengkapnya data yang dimasukkan sehingga gagal dalam menjalani *task scenario*, pada *task 8* terjadi kegagalan karena tidak cukupnya memori penyimpanan saat melakukan aktivitas unduh data pada *device* pengguna, kemudian pada *task 11* juga terjadi kegagalan dikarenakan terputusnya koneksi internet sehingga menyebabkan tidak keluarnya data yang diminta oleh pengguna.

Hasil rata-rata nilai pada *task scenario* menghasilkan data sebagai berikut:

Tabel 6.43 Rata-Rata Hasil Task Scenario

No	Pengguna	Rata-rata
1	Siswa	96,7 %
2	Pengajar	94,6 %

Untuk membatasi terjadinya kegagalan yang dapat terjadi pada pengguna, setiap aktivitas yang memerlukan koneksi internet, data yang lengkap dan kecukupan memori penyimpanan pada masing-masing *device* pengguna yang dapat terjadi maka diberikan pesan peringatan pada setiap kasus kegagalan yang kemungkinan dapat terjadi agar pengguna mengerti kesalahan yang terjadi.

6.3.3.2 Analisis Hasil Pengujian Usability Menggunakan USE Quissionaire

Pengujian dengan menggunakan kuesioner USE ini memiliki tujuan untuk mengetahui sejauh mana mana tingkat kegunaan, kemudahan dan kepuasan pengguna dalam menggunakan aplikasi yang sedang dikembangkan. Hasil pengujian aplikasi E-learning SMANTIGDA dengan menggunakan kuesioner USE memberikan hasil kelayakan yang cukup tinggi terutama pada beberapa pertanyaan dalam kategori *usefulness*, *ease of use* dan *ease of learning*. Hasil pengujian usability menggunakan kuesioner USE dapat dilihat pada Tabel 6.45.

Tabel 6.44 Presentasi Kelayakan Hasil Kuesioner USE

USEFULLNESS		KELAYAKAN
1	Aplikasi ini efektif untuk membantu kegiatan akademik maupun pembelajaran	100 %
2	Aplikasi ini membantu untuk lebih produktif lagi dalam kegiatan akademik maupun pembelajaran	100 %
3	Aplikasi ini berguna untuk kegiatan akademik sekolah dan pembelajaran untuk siswa dan pengajar	89 %
4	Aplikasi ini membantu menyelesaikan pekerjaan saya lebih mudah	89 %
5	Aplikasi ini menghemat waktu untuk menyelesaikan pekerjaan saya	88 %
6	Aplikasi ini memenuhi kebutuhan saya dalam hal kegiatan dan pembelajaran untuk sekolah	89 %
7	Aplikasi ini menjalankan fungsinya sesuai dengan ekspektasi saya	87 %
EASE OF USE		
8	Aplikasi ini mudah dimengerti	88 %
9	Hanya sedikit tahapan yang dibutuhkan dalam menggunakan	100 %

	aplikasi ini	
10	Dalam menggunakan aplikasi ini hanya membutuhkan sedikit usaha	85 %
11	Saya dapat menggunakan aplikasi ini tanpa petunjuk penggunaan	69 %
12	Aplikasi ini berfungsi dengan konsisten ketika digunakan	70 %
13	Saya bisa mengatasi kesalahan ketika menggunakan aplikasi ini dengan cepat dan mudah	81 %
14	Saya berhasil menggunakan aplikasi ini setiap saat saya mengaksesnya	87 %
15	Pengguna aplikasi Elearning maupun pengguna yang jarang menggunakan aplikasi Elearning akan menyukai aplikasi ini	87 %
EASE OF LEARNING		
16	Saya mempelajari cara penggunaan aplikasi dengan cepat	100 %
17	Saya dengan mudah mengingat bagaimana cara menggunakan aplikasi ini	89 %
18	Saya mudah dalam mempelajari cara menggunakan aplikasi ini	100 %
19	Saya dengan cepat dan terampil dalam menggunakan aplikasi ini	89 %
SATISFACTION		
20	Saya puas dengan aplikasi ini	78 %
21	Saya akan merekomendasikan aplikasi ini kepada orang lain	86 %
22	Sangat menyenangkan dalam menggunakan aplikasi ini	88 %
23	Aplikasi ini bekerja sesuai dengan keinginan saya	81 %
24	Saya nyaman dalam menggunakan aplikasi ini	72 %
25	Saya harus memiliki aplikasi ini	88 %

Berdasarkan Tabel 2.2 yang menunjukkan prosentase kecukupan pada kuesioner USE, didapati hasil rata-rata seperti pada Tabel 6.46, hasil rata-rata pengujian pada aplikasi E-learning SMANTIGDA memperoleh nilai pada masing-masing kriteria sebagai berikut:

Tabel 6.45 Rata-Rata Hasil Kuesioner USE

No	Aspek	Rata-rata
1	Usefulness	91,71 %
2	Ease of Use	82,37 %
3	Ease of Learning	94,5 %
4	Satisfaction	82,16 %

Kelayakan yang ditunjukkan pada kriteria *usefulness*, *ease of use*, *ease of learning* dan *satisfaction* yang dikembangkan memiliki nilai *range* antara 80% hingga 100%. Menurut tabel skala Likert hasil nilai yang didapat menunjukkan bahwa hasil dari pengujian usability menggunakan kuesioner USE dapat dikatakan sangat baik.

6.3.4 Analisis Hasil Pengujian *Compatibility*

Analisis hasil dari pengujian *compatibility* dilakukan dengan menjalankan aplikasi dari berbagai versi Android dari Android versi 5 (*Lollipop*) hingga Android versi terbaru dan diuji pada beberapa jenis *hardware*, jika aplikasi yang dikembangkan dapat dijalankan dengan baik maka hasil tersebut *valid*, semakin banyak perangkat dengan berbagai spesifikasi yang dicoba maka semakin baik aspek kompatibilitasnya (Liu, et al., 2014). Pengujian *compability* diuji pada tiga jenis perangkat yang berbeda, berdasarkan pernyataan yang telah dijelaskan sebelumnya dapat disimpulkan bahwa hasil pengujian *compability* E-Learning SMANTIGDA telah terpenuhi karena telah menghasilkan hasil uji yang *valid*.

BAB 7 PENUTUP

Pada bagian ini merupakan bagian akhir dari laporan penelitian. Bagian ini berisi kesimpulan yang akan menjawab rumusan masalah yang telah dipaparkan sebelumnya pada bab pendahuluan, serta saran dari penulis yang dapat digunakan untuk pengembangan aplikasi lebih lanjut.

7.1 Kesimpulan

Berdasarkan analisis yang dilakukan peneliti pada penelitian ini, maka dapat ditarik kesimpulan diantaranya:

1. Pengembangan aplikasi *e-learning* berbasis Android dengan menerapkan pendekatan *User Centered Design* terdapat beberapa tahapan:
 - a. Analisis kebutuhan: dengan menerapkan pendekatan *user centered design* pada tahap ini, dengan memahami konteks pengguna, menentukan pengguna dan kebutuhannya, dapat disimpulkan bahwa terdapat dua aktor pengguna aplikasi yang ditetapkan yaitu aktor pengguna siswa dan pengajar, serta mendapatkan 12 daftar kebutuhan untuk pengguna pengajar dan 11 daftar kebutuhan siswa setelah terjadinya dua kali iterasi pada tahap ini. Pada iterasi pertama terdapat beberapa perubahan kebutuhan yaitu dihapusnya kebutuhan registrasi, pergantian nama fungsi kritik saran menjadi aspirasi, terdapat fitur baru pada sisi siswa yaitu fitur unggah tugas dan penambahan fitur daftar akun siswa pada sisi pengajar. Kemudian pada iterasi kedua terdapat penambahan fungsi baru seperti info perubahan jadwal pada kedua sisi aplikasi, fungsi tambah kegiatan untuk sisi pengajar, fungsi *update* kata sandi dan fungsi unduh tugas pada sisi pengajar, serta terdapat pergantian nama fungsi pada fungsi kalender akademik menjadi *timeline* akademik
 - b. Perancangan: pada tahap ini terdapat perancangan arsitektur sistem dengan menetapkan minimum versi SDK 21 untuk implementasi sistemnya, serta menggunakan 3rd party API OneSignal, okHttp dan android-pdf-viewer, dan menetapkan Firebase dan SQLite sebagai basis datanya. Untuk perancangan basis data terdapat 7 perancangan basis data yaitu basis data akun siswa, aspirasi, info jadwal, *timeline* akademik, berkas tugas, tugas siswa dan latihan soal. Perancangan *class* diagram terdapat 14 *package class*, perancangan *sequence* diagram, perancangan algoritme dan terdapat 24 perancangan antarmuka untuk semua fitur aplikasi. Dengan menerapkan pendekatan UCD yaitu memberikan solusi dengan memberikan desain perancangan dan melakukan evaluasi secara langsung oleh pengguna, pada tahap ini terjadi 2 kali iterasi.

- c. Implementasi: pada tahap ini telah ditetapkan bahwa aplikasi ini diimplementasikan pada perangkat bergerak dengan sistem operasi Android API 21 keatas dengan menggunakan bahasa pemrograman Java serta menggunakan Firebase dan SQLite sebagai basis datanya.
 - d. Pengujian: yang dilakukan pada tahap ini yaitu melakukan pengujian fungsional dan non-fungsional, untuk pengujian fungsional terdapat pengujian unit yang menguji 5 unit *method* dengan hasil yang telah memiliki kesesuaian dengan tahap perancangan yang dilakukan sebelumnya, untuk pengujian validasi yang menguji seluruh kebutuhan yang ada pada sistem dengan hasil valid untuk seluruh kebutuhan, kemudian untuk pengujian non-fungsional terdapat pengujian *usability* dengan menggunakan kuesioner USE dan *task scenarios* yang diujikan langsung kepada 14 pengguna siswa dan 4 pengguna pengajar, serta pengujian *compatibility* yang dilakukan pada 3 *device* pengguna dengan spesifikasi yang berbeda.
2. Pengembangan aplikasi *e-learning* berbasis Android pada pengujian *usability* menggunakan Task Scenarios dengan tipe *full scale task scenarios* dan kuesioner USE untuk mengetahui empat aspek uji yaitu *usefulness*, *ease of use*, *ease of learning*, dan *satisfaction*. Dari kedua jenis pengujian *usability* tersebut didapati 2 kegagalan dari 13 *task scenario* untuk siswa dan 3 kegagalan dari 14 *task scenario* untuk pengajar yang disebabkan oleh tidak adanya koneksi internet, kosongnya data masukan, tidak ada kecocokan data dan penuhnya memori penyimpanan saat menjalankan aplikasi yang dikembangkan. Untuk rata-rata hasil nilai uji *usability* pada *task scenario* didapati 96,7% keberhasilan pada pengguna siswa dan 94,6% keberhasilan pada pengguna pengajar. Sedangkan untuk pengujian menggunakan kuesioner USE didapati rentang rata-rata hasil *usefulness* sebesar 91,71%, *ease of use* sebesar 83,37%, *ease of learning* sebesar 94,5%, dan *satisfaction* sebesar 82,16%. Dari nilai yang dihasilkan dari pengujian tersebut, dengan mengacu pada tabel skala Likert dengan hasil nilai *range* antara 80% hingga 100%, hasil dari pengujian *usability* menggunakan kuesioner USE dapat dikatakan mendapatkan nilai sangat baik.

7.2 Saran

Berdasarkan hasil penelitian yang dilakukan, penulis dapat menyampaikan saran untuk penelitian selanjutnya diantaranya sebagai berikut:

1. *User Interface* aplikasi kedua sisi dapat dibuat lebih sederhana dan ringan agar tetap nyaman digunakan pada oleh pengguna dengan spesifikasi *device* yang rendah.
2. Pada fungsi tugas diberikan fitur *reminder* sebelum batas waktu pengumpulan tugas.
3. Untuk menghindari besarnya *file size* aplikasi, diharapkan pengguna dapat melakukan akses dan menyimpan data apapun secara *online*.

Selain itu juga dapat melakukan pemisahan aplikasi siswa dan pengajar menjadi dua bagian aplikasi yang berbeda.



DAFTAR PUSTAKA

- A. S., R. & Shalahuddin, M., 2011. *Modul Pembelajaran Rekayasa Perangkat Lunak*). Bandung: Modula.
- Amadyah, A., 2008. Rancangan Sistem Pameran Online Menggunakan Metode UCD (User Centered Learning). *SNTS AKPRIND*.
- Barnum, C. M., 2011. Usability Testing Essentials: Ready, Set....Test!. 54(3), pp. 336 - 337.
- Bertram, D., n.d. *Libraries and Curtular Resources - University of Calgary*. [Online] Available at: pages.cpsc.ucalgary.ca/~saul/wiki/.../topic-dane-likert.doc [Accessed 3 July 2018].
- Devi, K. R., Sen, A. & K., H., 2012. A Working Framework for the User-Centered Design Approach and a Survey of the available Methods. *International Journal of Schientific and Research Publication*, 2(4).
- Gumelar, S., 2017. *PHP Indonesia Chapter Jakarta*. [Online] Available at: <https://phpid-jakarta.github.io/kulwap/2017/09/30/onesignal-push-notification-sasono-gumelar.html> [Accessed 18 05 2018].
- Jaya Kumar C, K., 2002. *Aplikasi 'E-Learning' Dalam Pengajaran Dan Pembelajaran Di Sekolah-Sekolah Malaysia: Cadangan Perlaksanaan Pada Senario Masa Kini*. Malaysia: Pasukan Projek Rintis Sekolah Bestari Bahagian Teknologi Pendidikan.
- Khalkhali, I. E., 2014. The use of DOKEOS e-learning platform in a Moroccan Business School. *International Conference on Multimedia Computing and Systems (ICMCS)*, pp. 633 - 638.
- KhedKar, S. & Thube, S., 2017. Real Time Databases for Application. *International Research Journal of Engineering and Technology*, 04(06), pp. 2078 - 2082.
- Kreibich, J. A., 2010. *Using SQLite*. 1st ed. Sebastopol, California: O'Reilly Media, Inc..
- Liu, J., Men, H. & Han, J., 2009. Comparative Study of Open-source E-learning Management Platform. *International Conference on Computational Intelligence and Software Engineering*, pp. 1 - 4.
- Liu, Z., Hu, Y. & Cai, L., 2014. Research on software security and compatibility test for mobile application.
- Lund, A. M., 2016. Measuring Usability with the USE Questionnaire.
- Margono, 2004. *Metodologi Penelitian Pendidikan*. Jakarta: PT. Rineka Cipta.
- Mayers, G. J., 2004. *The Art of Software Testing, Second Edition*. Canada: John Wiley & Sons, Inc., Hoboken, Ney Jersey.

OneSignal, I., 2018. *OneSignal*. [Online]
Available at: <https://onesignal.com/>
[Accessed 18 05 2018].

Pramita, D. A., 2015. Rancang Bangun Push Notification Berbasis Moodle pada Perangkat Android. pp. 99-101.

Pratama, A. R., 2016. *CODEPOLITAN*. [Online]
Available at: <https://www.codepolitan.com/belajar-uml-sequence-diagram-57fdb1a5ba777-17044>
[Accessed 9 July 2018].

Preece, J., 1994. *Human - Computer Interaction*. Addison: Wesley.

Presman, R. S., 1997. *Software Engineering*. 4 ed. s.l.: Mc. Graw Hill.

Pusuluri, N. R., 2006. *Software Testing Concepts And Tools*. s.l.:Dreamtech Press.

Putra, A. S. P., 2013. Rancang Banung E-Learning Berbasis Moodle Menggunakan Aplikasi Android. pp. 87-88.

Ramdhan, I. C., 2011. Pengembangan Aplikasi E-Learning Berbasis Moodle (Studi Kasus : SMA 2 Mei Ciputat).

Rianto Rahadi, D., 2014. Pengukuran Usability Sistem Menggunakan Use Questionnaire Pada Aplikasi Android. *Jurnal Sistem Informasi (JSI)*, 6(1), pp. 661-671.

Rosenberg, M. J., 2001. *Strategies for delivering knowledge in the digital age*. New York: McGraw-Hill Companies, Inc..

Rubin, J. & Chrisnell, D., 2008. *Handbook Of Usability Testing*. 2nd ed. Indianapolis: Wiley Publishing.Inc.

Soekartawi, 2002. *Prospek Pembelajaran Melalui Internet*. Jakarta, UT-Pustekkom dan IPTPI.

STAPIĆ, Z., 2013. *A PROPOSAL OF AN ONTOLOGY-BASED METHODOLOGICAL FRAMEWORK FOR MULTI-PLATFORM MOBILE APPLICATIONS DEVELOPMENT*. Alcalá de Henares (Madrid): University of Alcalá Computer Science Department, Postgraduate School Doctoral program "Information and Knowledge Engineering".

Statista, 2017. *The Statistic Portal*. [Online]
Available at: <https://www.statista.com/>
[Accessed 2018].

ST, H., 2015. *Cepat Mahir MySQL dan SQLite*. s.l.:Elex Media Komputindo.

Triadi, D., 2013. *Bedah Tuntas Fitur Android*. 1 ed. Yogyakarta: Jogja Great! Publisher.

Widhiarso, W., J. & S., 2007. Metode UCD (User Centered Design) Untuk Rancangan Kios Informasi Studi Kasus : Rumah Sakit Bersalin XYZ. *STMIK MDP Palembang*.